

# RA8E1 Group

User's Manual: Hardware

## 32-Bit MCU

Renesas Advanced (RA) Family

Renesas RA8 Series

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

## Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
  2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
  3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
  4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
  5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
  6. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The intended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below.
    - "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
    - "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
- Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user's manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION ("Vulnerability Issues"). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
  8. When using Renesas Electronics products, refer to the latest product information (data sheets, user's manuals, application notes, "General Notes for Handling and Using Semiconductor Devices" in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
  9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
  10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
  11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
  12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
  13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
  14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,  
Koto-ku, Tokyo 135-0061, Japan  
[www.renesas.com](http://www.renesas.com)

## Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

## Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:  
[www.renesas.com/contact/](http://www.renesas.com/contact/)

# General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

## 1. Precaution against Electrostatic Discharge (ESD)

A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity.

Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.

## 2. Processing at power-on

The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.

## 3. Input of signal during power-off state

Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.

## 4. Handling of unused pins

Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.

## 5. Clock signals

After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.

## 6. Voltage application waveform at input pin

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between  $V_{IL}$  (Max.) and  $V_{IH}$  (Min.).

## 7. Prohibition of access to reserved addresses

Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.

## 8. Differences between products

Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

# Preface

## 1. About this document

This manual is generally organized into an overview of the product, descriptions of the CPU, system control functions, peripheral functions, electrical characteristics, and usage notes. This manual describes the product specification of the microcontroller (MCU) superset. Depending on your product, some pins, registers, or functions might not exist. Address space that store unavailable registers are reserved.

## 2. Audience

This manual is written for system designers who are designing and programming applications using the Renesas Microcontroller. The user is expected to have basic knowledge of electrical circuits, logic circuits, and the MCU.

## 3. Renesas Publications

Renesas provides the following documents. Before using any of these documents, visit [www.renesas.com](http://www.renesas.com) for the most up-to-date version of the document.

Component	Document Type	Description
Microcontrollers	Data sheet	Features, overview, and electrical characteristics of the MCU
	User's Manual: Hardware	MCU specifications such as pin assignments, memory maps, peripheral functions, electrical characteristics, timing diagrams, and operation descriptions
	Application Notes	Technical notes, board design guidelines, and software migration information
	Technical Update (TU)	Preliminary reports on product specifications such as restriction and errata
Software	User's Manual: Software	API reference and programming information
	Application Notes	Project files, guidelines for software programming, and application examples to develop embedded software applications
Tools & Kits, Solutions	User's Manual: Development Tools	User's manual and quick start guide for developing embedded software applications with Development Kits (DK), Starter Kits (SK), Promotion Kits (PK), Product Examples (PE), and Application Examples (AE)
	User's Manual: Software	
	Quick Start Guide	
	Application Notes	Project files, guidelines for software programming, and application examples to develop embedded software applications

## 4. Numbering Notation

The following numbering notation is used throughout this manual:

Example	Description
011b	Binary number. For example, the binary equivalent of the number 3 is 011b.
0x1F	Hexadecimal number. For example, the hexadecimal equivalent of the number 31 is described 0x1F. In some cases, a hexadecimal number is shown with the suffix "h".
1234	Decimal number. A decimal number is followed by this symbol only when the possibility of confusion exists. Decimal numbers are generally shown without a suffix.

## 5. Typographic Notation

The following typographic notation is used throughout this manual:

Example	Description
AAA.BBB.CCC	Periods separated a function module symbol (AAA), register symbol (BBB), and bit field symbol (CCC).
AAA.BBB	A period separated a function module symbol (AAA) and register symbol (BBB).
BBB.DDD	A period separated a register symbol (BBB) and bit field symbol (DDD).
EEE[3:0]	Numbers in brackets expresses a bit number. For example, EEE[3:0] occupies bits 3 to 0.

## 6. Unit and Unit Prefix

The following units and unit prefixes are sometimes misleading. Those unit prefixes are described throughout this manual with the following meaning:

Symbol	Name	Description
b	Binary Digit	Single 0 or 1
B	Byte	This unit is generally used for memory specification of the MCU and address space.
k	kilo-	$1000 = 10^3$ . k is also used to denote 1024 ( $2^{10}$ ) but this unit prefix is used to denote 1000 ( $10^3$ ) throughout this manual.
K	Kilo-	$1024 = 2^{10}$ . This unit prefix is used to denote 1024 ( $2^{10}$ ) not 1000 ( $10^3$ ) throughout this manual.

## 7. Special Terms

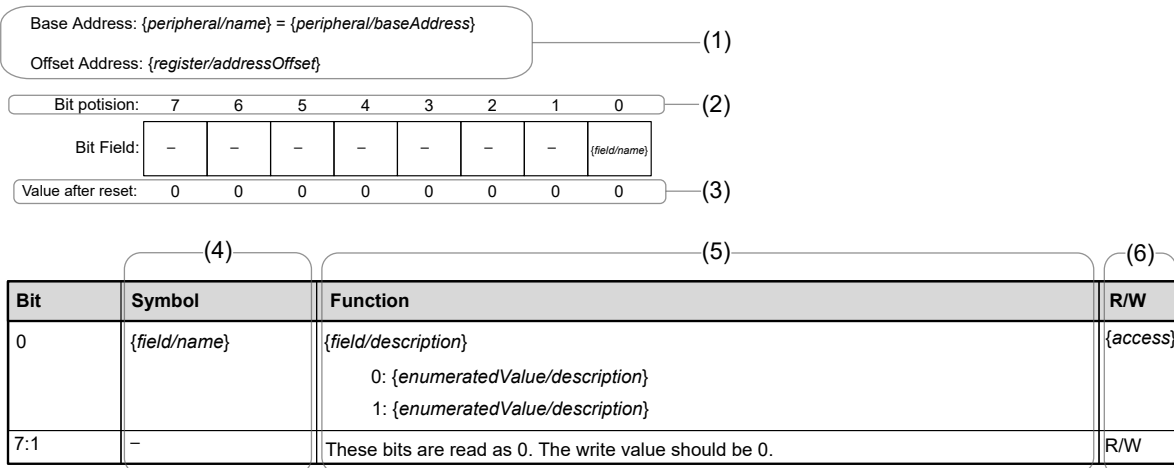
The following terms have special meanings.

Term	Description
NC	Not connected pin. This pin should be left floating unless specified otherwise.
Hi-Z	High impedance.
x	Don't care or undefined.

## 8. Register Description

Each register description includes both a register diagram that shows the bit assignments and a register bit table that describes the content of each bit. The example of symbols used in these tables are described in the sections that follow. The following is an example of a register description and associated bit field definition.

XX.X.X {register/name} : {register/description}



### (1) Function module symbol, register symbol, and address assignment

Function module symbol, {peripheral/name}, register symbol, {register/name}, and address assignment of this register are generally expressed. Base Address and Offset Address mean {register/name} : {register/description} of {peripheral/name} is assigned to address {peripheral/baseAddress} + {register/addressOffset}.

### (2) Bit number

This number indicates the bit number. This bits are shown in order from bits 31 to 0 for 32-bit register, from bits 15 to 0 for 16-bit register, and from bits 7 to 0 for 8-bit register.

### (3) Value after reset

This symbol or number indicate the value of each bit after a hard reset. The value is shown in binary unless specified otherwise.

- 0: Indicates that the value is 0 after a reset.
- 1: Indicates that the value is 1 after a reset.
- x: Indicates that the value is undefined after a reset.

### (4) Symbol

{field/name} indicates the short name of bit field. Reserved bit is expressed with a —.

### (5) Function

Function indicates the full name of the bit field, {field/description}, and enumerated values.

### (6) R/W

The R/W column indicates access type whether the bit field is readable or writable.

- R/W: The bit field is readable and writable.
- R: The bit field is readable only. Writing to this bit field has no effect.
- W: The bit field is writable only. The read value is the same as after a reset unless specified otherwise.

## 9. Abbreviations

Abbreviations used in this document are shown in the following table.

Abbreviation	Description
AES	Advanced Encryption Standard
AHB	Advanced High-performance Bus
AHB-AP	AHB Access Port
APB	Advanced Peripheral Bus
ARC	Alleged RC
ATB	Advanced Trace Bus
BCD	Binary Coded Decimal
BSDL	Boundary Scan Description Language
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
ETB	Embedded Trace Buffer
ETM	Embedded Trace Macrocell
FLL	Frequency Locked Loop
FPU	Floating Point Unit
HMI	Human Machine Interface
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NVIC	Nested Vector Interrupt Controller
PC	Program Counter
PFS	Port Function Select
PLL	Phase Locked Loop
POR	Power-on reset
PWM	Pulse Width Modulation
RSA	Rivest Shamir Adleman
SHA	Secure Hash Algorithm
S/H	Sample and Hold
SP	Stack Pointer
SWD	Serial Wire Debug
SW-DP	Serial Wire-Debug Port
TRNG	True Random Number Generator
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

## 10. Proprietary Notice

All text, graphics, photographs, trademarks, logos, artwork and computer code, collectively known as content, contained in this document is owned, controlled or licensed by or to Renesas, and is protected by trade dress, copyright, patent and trademark laws, and other intellectual property rights and unfair competition laws. Except as expressly provided herein, no part of this document or content may be copied, reproduced, republished, posted, publicly displayed, encoded, translated, transmitted or distributed in any other medium for publication or distribution or for any commercial enterprise, without prior written consent from Renesas.

Arm® and Cortex® are registered trademarks of Arm Limited. CoreSight™ is a trademark of Arm Limited.

CoreMark® is a registered trademark of the Embedded Microprocessor Benchmark Consortium.

Magic Packet™ is a trademark of Advanced Micro Devices, Inc.

Other brands and names mentioned in this document may be the trademarks or registered trademarks of their respective holders.

## 11. Feedback on the product

If you have any comments or suggestions about this product, go to [Contact Us](#).



# Contents

<b>Features</b> .....	<b>56</b>
<b>1. Overview</b> .....	<b>57</b>
1.1 Function Outline .....	57
1.2 Block Diagram .....	62
1.3 Part Numbering .....	62
1.4 Function Comparison .....	64
1.5 Pin Functions.....	65
1.6 Pin Assignments.....	70
1.7 Pin Lists .....	72
<b>2. CPU</b> .....	<b>76</b>
2.1 Overview.....	76
2.1.1 CPU.....	76
2.1.2 Debug.....	77
2.2 Block Diagram .....	77
2.3 Implementation Options.....	78
2.4 Trace Interface.....	79
2.5 JTAG/SWD Interface .....	79
2.6 Initial Vector Table Base Address .....	80
2.7 Debug Function .....	80
2.7.1 External Debugger Connectivity and Authentication Through JTAG/SWD .....	80
2.7.2 Debug Protection Mechanism by Software .....	81
2.7.3 Effect of Debug Function.....	82
2.8 Registers Description .....	83
2.8.1 CPU Control Registers .....	83
2.8.2 Address Spaces for External Debugger .....	85
2.8.3 CPU Peripheral .....	86
2.8.4 ROM Table .....	87
2.8.5 DBGREG.....	89
2.8.6 OCDREG .....	96
2.9 CoreSight Cross Trigger Interface (CTI).....	102
2.10 CoreSight ATB Funnel.....	102
2.11 SysTick System Timer .....	103
2.12 CoreSight Timestamp Generator.....	104
2.13 OCD Emulator Connection .....	104
2.13.1 SYOCDR.DBGEN .....	106
2.13.2 Restriction in OCD Emulator Connecting.....	106
2.13.3 Starting Connection While in Low Power Mode .....	106

2.13.4	Low Power Mode Change During Debugging .....	106
2.13.5	Connecting Sequence and JTAG/SWD Authentication .....	107
2.14	References .....	107
2.15	Usage Notes .....	108
2.15.1	Restrictions for OSPI .....	108
<b>3.</b>	<b>Operating Modes .....</b>	<b>109</b>
3.1	Overview .....	109
3.2	Operating Mode Types and Selection .....	109
3.3	Details of Operating Modes .....	109
3.3.1	Single-Chip Mode .....	109
3.3.2	JTAG Boot Mode .....	109
3.3.3	SCI Boot Mode .....	109
3.3.4	USB Boot Mode .....	109
3.4	Operating Modes Transitions .....	109
3.4.1	Operation modes and the relationship of Mode Transition .....	109
<b>4.</b>	<b>Address Space .....</b>	<b>111</b>
4.1	Address Space .....	111
4.2	External Address Space .....	113
<b>5.</b>	<b>Resets .....</b>	<b>115</b>
5.1	Overview .....	115
5.2	Register Descriptions .....	121
5.2.1	RSTSAR : Reset Security Attribution Register .....	121
5.2.2	RSTSR0 : Reset Status Register 0 .....	122
5.2.3	RSTSR1 : Reset Status Register 1 .....	124
5.2.4	RSTSR2 : Reset Status Register 2 .....	126
5.2.5	SYRSTMSK0 : System Reset Mask Control Register 0 .....	126
5.2.6	SYRSTMSK2 : System Reset Mask Control Register 2 .....	127
5.3	Operation .....	127
5.3.1	RES Pin Reset .....	127
5.3.2	Power-On Reset .....	127
5.3.3	Voltage Monitor Reset .....	128
5.3.4	Deep Software Standby Reset .....	131
5.3.5	Independent Watchdog Timer Reset .....	131
5.3.6	Watchdog Timer Reset .....	132
5.3.7	CPU Lockup Reset .....	132
5.3.8	Software Reset .....	132
5.3.9	Bus Error Reset .....	132
5.3.10	Common Memory Error Reset .....	132
5.3.11	Determination of Cold/Warm Start .....	133

5.3.12	Determination of Reset Generation Source .....	133
5.4	Reset Protection from Non-secure Programs.....	134
<b>6.</b>	<b>Option-Setting Memory .....</b>	<b>135</b>
6.1	Overview.....	135
6.2	Register Descriptions .....	136
6.2.1	OFS0 : Option Function Select Register 0 .....	136
6.2.2	OFS2 : Option Function Select Register 2 .....	139
6.2.3	DUALSEL : Dual Mode Select Register .....	139
6.2.4	SAS : Startup Area Setting Register .....	140
6.2.5	OFS1, OFS1_SEC : Option Function Select Register 1 for Non-secure and Secure .....	141
6.2.6	OFS1_SEL : Option Function Select Register 1 for Security Attribution.....	142
6.2.7	BANKSEL, BANKSEL_SEC : Bank Select Register for Non-secure and Secure.....	143
6.2.8	BANKSEL_SEL : Bank Select Register for Security Attribution.....	145
6.2.9	BPS, BPS_SEC, BPS_SEL : Block Protect Setting Register.....	146
6.2.10	PBPS, PBPS_SEC : Permanent Block Protect Setting Register .....	148
6.2.11	CFGD0LOCK : Configuration Data 0 Lock Bit .....	148
6.2.12	CFGD1LOCK : Configuration Data 1 Lock Bit .....	149
6.2.13	CFGD2LOCK : Configuration Data 2 Lock Bit .....	150
6.2.14	ARCLS : Anti-Rollback Counter Lock Setting .....	151
6.2.15	ARCCS : Anti-Rollback Counter Configuration Setting for Non-secure Application .....	151
6.2.16	ARC_SECn : Anti-Rollback Counter for Secure Application n (n = 0, 1) .....	152
6.2.17	ARC_NSECn : Anti-Rollback Counter for Non-secure Application n (n = 0 to 7) .....	152
6.3	Setting Option-Setting Memory .....	152
6.3.1	Allocation of Data in Option-Setting Memory .....	152
6.3.2	Setting Data for Programming Option-Setting Memory.....	152
6.3.3	Security attribution of code flash option-setting memory .....	153
6.3.4	Timing of the Setting Value .....	153
6.4	Usage Notes.....	153
6.4.1	Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory	153
<b>7.</b>	<b>Programmable Voltage Detection (PVD) .....</b>	<b>155</b>
7.1	Overview.....	155
7.2	Register Descriptions .....	156
7.2.1	PVDSAR : Programable Voltage Detection Security Attribution Register .....	156
7.2.2	PVDmCMPCR : Voltage Monitor m Comparator Control Register (m = 1, 2) .....	157
7.2.3	PVDmCR0 : Voltage Monitor m Circuit Control Register 0 (m = 1, 2).....	158
7.2.4	PVDmCR1 : Voltage Monitor m Circuit Control Register 1 (m = 1, 2).....	159
7.2.5	PVDmSR : Voltage Monitor m Circuit Status Register (m = 1, 2).....	160
7.2.6	PVDmFCR : Voltage Monitor m Function Control Register (m = 1, 2) .....	160
7.3	VCC Input Voltage Monitor .....	161

7.3.1	Monitoring $V_{det0}$ .....	161
7.3.2	Monitoring $V_{detm}$ .....	161
7.4	Reset from Voltage Monitor 0 .....	161
7.5	Interrupt and Reset from Voltage Monitor m (m = 1, 2) .....	162
7.6	Event Link Controller (ELC) Output .....	165
7.6.1	Interrupt Handling and Event Linking .....	166
<b>8.</b>	<b>Clock Generation Circuit .....</b>	<b>167</b>
8.1	Overview.....	167
8.2	Register Descriptions .....	171
8.2.1	CGFSAR : Clock Generation Function Security Attribute Register .....	171
8.2.2	SCKDIVCR : System Clock Division Control Register .....	174
8.2.3	SCKDIVCR2 : System Clock Division Control Register 2 .....	176
8.2.4	SYRACCR : System Register Access Control Register.....	176
8.2.5	SCKSCR : System Clock Source Control Register .....	177
8.2.6	PLLCCR : PLL Clock Control Register.....	178
8.2.7	PLLCCR2 : PLL Clock Control Register 2.....	179
8.2.8	PLLCR : PLL Control Register .....	180
8.2.9	PLL2CCR : PLL2 Clock Control Register.....	181
8.2.10	PLL2CCR2 : PLL2 Clock Control Register 2.....	182
8.2.11	PLL2CR : PLL2 Control Register .....	183
8.2.12	MOSCCR : Main Clock Oscillator Control Register .....	183
8.2.13	SOSCCR : Sub-Clock Oscillator Control Register .....	184
8.2.14	LOCOCR : Low-Speed On-Chip Oscillator Control Register .....	185
8.2.15	HOCOOCR : High-Speed On-Chip Oscillator Control Register .....	186
8.2.16	HOCOOCR2 : High-Speed On-Chip Oscillator Control Register2 .....	187
8.2.17	MOCOOCR : Middle-Speed On-Chip Oscillator Control Register .....	187
8.2.18	FLLCR1 : FLL Control Register1.....	188
8.2.19	FLLCR2 : FLL Control Register2.....	189
8.2.20	OSCSF : Oscillation Stabilization Flag Register .....	190
8.2.21	OSCMONR : Oscillator Monitor Register .....	191
8.2.22	OSTDCR : Oscillation Stop Detection Control Register.....	192
8.2.23	OSTDSR : Oscillation Stop Detection Status Register .....	193
8.2.24	MOSCWTCR : Main Clock Oscillator Wait Control Register.....	194
8.2.25	MOMCR : Main Clock Oscillator Mode Oscillation Control Register.....	194
8.2.26	SOMCR : Sub-Clock Oscillator Mode Control Register .....	195
8.2.27	CKOOCR : Clock Out Control Register .....	196
8.2.28	LOCOUTCR : LOCO User Trimming Control Register .....	197
8.2.29	MOCOUTCR : MOCO User Trimming Control Register .....	197
8.2.30	HOCOUTCR : HOCO User Trimming Control Register .....	198
8.2.31	MOSCSCR : Main Clock Oscillator Standby Control Register.....	198

8.2.32	HOCOSCR : High-Speed On-Chip Oscillator Standby Control Register .....	199
8.2.33	USBCKDIVCR : USB Clock Division Control Register.....	199
8.2.34	OCTACKDIVCR : Octal-SPI Clock Division Control Register .....	200
8.2.35	CANFDCKDIVCR : CANFD Core Clock Division Control Register.....	201
8.2.36	USBCKCR : USB Clock Control Register .....	201
8.2.37	OCTACKCR : Octal-SPI Clock Control Register .....	202
8.2.38	CANFDCKCR : CANFD Core Clock Control Register .....	204
8.2.39	SCICKDIVCR : SCI clock Division control register .....	205
8.2.40	SPICKDIVCR : SPI clock Division control register .....	205
8.2.41	SCICKCR : SCI clock control register .....	206
8.2.42	SPICKCR : SPI clock control register .....	207
8.2.43	TRCKCR : Trace Clock Control Register .....	208
8.3	Main Clock Oscillator.....	209
8.3.1	Connecting a Crystal Resonator .....	209
8.3.2	External Clock Input.....	210
8.3.3	Notes on External Clock Input.....	210
8.3.4	Note on Oscillation Keep in Software Standby mode .....	211
8.4	Sub-Clock Oscillator .....	211
8.4.1	Connecting a 32.768-kHz Crystal Resonator.....	211
8.4.2	Pin Handling When the Sub-Clock Oscillator Is Not Used .....	212
8.4.3	External Clock Input.....	212
8.5	Low speed On Chip Oscillator (LOCO) .....	212
8.5.1	LOCO Operating conditions .....	212
8.5.2	LOCO Stop conditions .....	213
8.5.3	Restrictions on LOCO operation and stopping.....	213
8.6	Middle speed On Chip Oscillator (MOCO) .....	214
8.6.1	MOCO Operating conditions.....	214
8.6.2	MOCO Stop conditions .....	214
8.6.3	Restrictions on MOCO operation and stopping.....	214
8.7	High speed On Chip Oscillator (HOCO).....	215
8.7.1	Frequency Locked Loop Function.....	215
8.7.2	Note on Oscillation Keep in Software Standby mode .....	215
8.7.3	Note on using as TRACE clock source .....	216
8.8	PLL Circuit.....	216
8.9	Oscillation Stop Detection Function.....	217
8.9.1	Oscillation Stop Detection and Operation after Detection.....	217
8.9.2	Oscillation Stop Detection Interrupts.....	219
8.10	Internal Clock.....	220
8.10.1	CPU Clock (CPUCLK).....	220
8.10.2	System Clock (ICLK).....	221

8.10.3	Debugger Clock (DCLK) .....	222
8.10.4	Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE) .....	222
8.10.5	FlashIF Clock (FCLK).....	222
8.10.6	Trace Clock (TRCLK).....	223
8.10.7	SCI Clock (SCICLK).....	223
8.10.8	SPI Clock (SPICLK) .....	223
8.10.9	Octal-SPI clock (OCTACLK, OCTADIVCLK).....	223
8.10.10	CANFD Core clock (CANFDCLK) .....	224
8.10.11	USB Clock (USBCLK) .....	224
8.10.12	CAN Clock (CANMCLK).....	224
8.10.13	ULPT Clock (ULPTLCLK, ULPTSCLK).....	224
8.10.14	CAC Clock (CACCLK).....	225
8.10.15	RTC Clock (RTCSCLK, RTCLCLK) .....	225
8.10.16	IWDT Clock (IWDTCLK) .....	225
8.10.17	AGT Clock (AGTSCLK, AGTLCLK) .....	225
8.10.18	SysTick Timer Clock (SYSTICKCLK).....	225
8.10.19	External Pin Output Clock (CLKOUT) .....	225
8.10.20	JTAG Clock (JTAGTCK).....	225
8.11	Clock Setting .....	225
8.11.1	System Clock Setting .....	225
8.11.2	Peripheral module-dedicated Clock Setting .....	231
8.11.3	Main Clock Oscillator Setting .....	233
8.11.4	Sub-Clock Oscillator Setting .....	234
8.11.5	High-speed On Chip Oscillator Setting .....	234
8.11.6	PLL Setting.....	236
8.12	Usage Notes.....	237
8.12.1	Notes on Clock Generation Circuit.....	237
8.12.2	Notes on Resonator .....	238
8.12.3	Notes on Board Design .....	238
8.12.4	Notes on Resonator Connect Pin.....	238
8.12.5	Notes on Using Sub-Clock Oscillator .....	238
<b>9.</b>	<b>Clock Frequency Accuracy Measurement Circuit (CAC).....</b>	<b>239</b>
9.1	Overview.....	239
9.2	Register Descriptions .....	240
9.2.1	CACR0 : CAC Control Register 0 .....	240
9.2.2	CACR1 : CAC Control Register 1 .....	241
9.2.3	CACR2 : CAC Control Register 2 .....	242
9.2.4	CAICR : CAC Interrupt Control Register .....	243
9.2.5	CASTR : CAC Status Register .....	244
9.2.6	CAULVR : CAC Upper-Limit Value Setting Register .....	245

9.2.7	CALLVR : CAC Lower-Limit Value Setting Register .....	245
9.2.8	CACNTBR : CAC Counter Buffer Register .....	245
9.3	Operation .....	246
9.3.1	Measuring Clock Frequency .....	246
9.3.2	Digital Filtering of Signals on CACREF Pin .....	247
9.4	Interrupt Requests .....	247
9.5	Usage Notes .....	247
9.5.1	Settings for the Module-Stop Function .....	247
<b>10.</b>	<b>Low Power Modes .....</b>	<b>248</b>
10.1	Overview .....	248
10.2	Register Descriptions .....	253
10.2.1	LPMSAR : Low Power Mode Security Attribution Register .....	253
10.2.2	DPFSAR : Deep Software Standby Interrupt Factor Security Attribution Register .....	254
10.2.3	RSCSAR : RAM Standby Control Security Attribution Register .....	257
10.2.4	MSTPCRA : Module Stop Control Register A .....	258
10.2.5	MSTPCRB : Module Stop Control Register B .....	258
10.2.6	MSTPCRC : Module Stop Control Register C .....	260
10.2.7	MSTPCRD : Module Stop Control Register D .....	261
10.2.8	MSTPCRE : Module Stop Control Register E .....	263
10.2.9	OPCCR : Operating Power Control Register .....	264
10.2.10	PDRAMSCR0 : SRAM Power Domain Standby Control Register 0 .....	265
10.2.11	PDRAMSCR1 : SRAM Power Domain Standby Control Register 1 .....	266
10.2.12	SSCR1 : Software Standby Control Register 1 .....	267
10.2.13	LPSCR : Low Power State Control Register .....	267
10.2.14	DPSBYCR : Deep Software Standby Control Register .....	268
10.2.15	DPSWCR : Deep Software Standby Wait Control Register .....	269
10.2.16	DPSIER0 : Deep Software Standby Interrupt Enable Register 0 .....	270
10.2.17	DPSIER1 : Deep Software Standby Interrupt Enable Register 1 .....	270
10.2.18	DPSIER2 : Deep Software Standby Interrupt Enable Register 2 .....	271
10.2.19	DPSIER3 : Deep Software Standby Interrupt Enable Register 3 .....	272
10.2.20	DPSIFR0 : Deep Software Standby Interrupt Flag Register 0 .....	273
10.2.21	DPSIFR1 : Deep Software Standby Interrupt Flag Register 1 .....	274
10.2.22	DPSIFR2 : Deep Software Standby Interrupt Flag Register 2 .....	275
10.2.23	DPSIFR3 : Deep Software Standby Interrupt Flag Register 3 .....	276
10.2.24	DPSIEGR0 : Deep Software Standby Interrupt Edge Register 0 .....	278
10.2.25	DPSIEGR1 : Deep Software Standby Interrupt Edge Register 1 .....	278
10.2.26	DPSIEGR2 : Deep Software Standby Interrupt Edge Register 2 .....	279
10.2.27	SYOCDRCR : System Control OCD Control Register .....	280
10.2.28	PLL1LDOCR : PLL1-LDO Control Register .....	280
10.2.29	PLL2LDOCR : PLL2-LDO Control Register .....	281

10.2.30	HOCOLDOCR : HOCO-LDO Control Register .....	282
10.3	Reducing Power Consumption by Switching Clock Signals .....	282
10.4	Module-Stop Function .....	282
10.5	Function for Lower Operating Power Consumption.....	284
10.5.1	Setting Operating Power Control Mode .....	284
10.6	Low Power Modes .....	284
10.6.1	CPU Sleep Mode .....	284
10.6.2	CPU Deep Sleep mode .....	286
10.6.3	Software Standby Mode.....	287
10.6.4	Deep Software Standby Mode .....	290
10.7	Usage Notes.....	293
10.7.1	Register Access .....	293
10.7.2	I/O Port States.....	295
10.7.3	Module-Stop State of DMAC and DTC .....	295
10.7.4	Internal Interrupt sources .....	295
10.7.5	Input Buffer Control by DIRQnE Bit.....	295
10.7.6	Transition to Low Power Modes .....	295
10.7.7	Timing of WFI Instructions.....	295
10.7.8	Writing WDT/IWDT registers by DMAC or DTC .....	296
10.7.9	Module stop bit write timing.....	296
10.7.10	Notes on transitioning to or canceling low power state .....	296
<b>11.</b>	<b>Battery Backup Function .....</b>	<b>299</b>
11.1	Overview.....	299
11.1.1	Battery backup function block diagram .....	299
11.1.2	Features of Battery Backup Function.....	300
11.1.3	Battery Power Supply Switch .....	301
11.1.4	VBATT_R voltage drop detection function .....	301
11.1.5	Backup Registers .....	301
11.1.6	Time Capture Function.....	301
11.1.7	Tamper Detection Function .....	301
11.1.8	VBATT Voltage Monitor Function .....	301
11.2	Register Descriptions .....	301
11.2.1	BBFSAR : Battery Backup Function Security Attribute Register.....	301
11.2.2	VBRABAR : VBATT Backup Register Security Attribute Boundary Address Register .....	302
11.2.3	VBRPABARS : VBATT Backup Register Privilege Attribute Boundary Address Register for Secure Region .....	303
11.2.4	VBRPABARNS : VBATT Backup Register Privilege Attribute Boundary Address Register for Non-secure Region .....	304
11.2.5	VBATTMNSLR : Battery Backup Voltage Monitor Function Select Register .....	304
11.2.6	VBTBER : VBATT Backup Enable Register.....	305



11.2.7	VBTBKR[n] : VBATT Backup Register (n = 0 to 127).....	305
11.2.8	VBTICTLR : VBATT Input Control Register.....	306
11.2.9	VBTICTLR2 : VBATT Input Control Register 2.....	307
11.2.10	VBTIMONR : VBATT Input Monitor Register.....	307
11.2.11	VBTBPCR1 : VBATT Battery Power Supply Control Register 1 .....	308
11.2.12	VBTBPCR2 : VBATT Battery Power Supply Control Register 2 .....	308
11.2.13	VBTBPSR : VBATT Battery Power Supply Status Register .....	309
11.2.14	VBTADSR : VBATT Tamper detection Status Register .....	310
11.2.15	VBTADCR1 : VBATT Tamper detection Control Register 1 .....	311
11.2.16	VBTADCR2 : VBATT Tamper detection Control Register 2 .....	312
11.3	Operation.....	312
11.3.1	Battery Backup Function .....	312
11.3.2	VBATT Battery Power Supply Switch Usage .....	314
11.3.3	VBATT_R voltage drop detection function Usage.....	315
11.3.4	VBATT Backup Register Usage .....	315
11.3.5	Tamper Detection Usage.....	315
11.3.6	VBATT voltage monitor function Usage .....	316
11.3.7	Initial Settings Examples .....	317
11.4	Interrupt Sources .....	318
11.5	Usage Notes.....	318
<b>12.</b>	<b>Register Write Protection .....</b>	<b>319</b>
12.1	Overview.....	319
12.2	Register Descriptions .....	319
12.2.1	PRCR_S : Protect Register for Secure (PRCR_S) .....	319
12.2.2	PRCR_NS : Protect Register for Non-secure (PRCR_NS).....	320
<b>13.</b>	<b>Interrupt Controller Unit (ICU).....</b>	<b>322</b>
13.1	Overview.....	322
13.2	Register Descriptions .....	323
13.2.1	ICUSARA : Interrupt Controller Unit Security Attribution Register A.....	323
13.2.2	ICUSARB : Interrupt Controller Unit Security Attribution Register B.....	324
13.2.3	ICUSARE : Interrupt Controller Unit Security Attribution Register E.....	325
13.2.4	ICUSARF : Interrupt Controller Unit Security Attribution Register F .....	326
13.2.5	ICUSARG : Interrupt Controller Unit Security Attribution Register G .....	327
13.2.6	ICUSARH : Interrupt Controller Unit Security Attribution Register H .....	327
13.2.7	ICUSARI : Interrupt Controller Unit Security Attribution Register I.....	328
13.2.8	TEVTRCR : Trusted Event Route Control Register .....	328
13.2.9	IRQCRi : IRQ Control Register i (i = 0 to 15) .....	329
13.2.10	NMISR : Non-Maskable Interrupt Status Register .....	330
13.2.11	NMIER : Non-Maskable Interrupt Enable Register .....	333

13.2.12	NMICLR : Non-Maskable Interrupt Status Clear Register.....	334
13.2.13	NMICR : NMI Pin Interrupt Control Register .....	335
13.2.14	IELSRn : ICU Event Link Setting Register n (n = 0 to 95).....	336
13.2.15	WUPEN0 : Wake Up Interrupt Enable Register 0 .....	338
13.2.16	WUPEN1 : Wake Up interrupt enable register 1 .....	340
13.3	Vector Table.....	342
13.3.1	Interrupt Vector Table .....	342
13.3.2	Event Number .....	345
13.4	Maskable Interrupt Operation .....	352
13.4.1	Detecting Interrupts.....	352
13.5	Maskable Interrupt setting procedure .....	353
13.5.1	Operations During an Interrupt.....	353
13.5.2	Enabling Interrupt Requests.....	353
13.5.3	Disabling Interrupt Requests.....	353
13.5.4	Polling for interrupts .....	354
13.5.5	Selecting Interrupt Request Destinations.....	354
13.5.6	Digital Filter .....	355
13.5.7	External Pin Interrupts.....	355
13.6	Non-Maskable Interrupt Operation .....	356
13.6.1	Correspondence to TrustZone-M by NMI.....	356
13.7	Security.....	357
13.7.1	Security Related to CPU Interrupt Inputs .....	357
13.7.2	Trusted Interrupt Management.....	358
13.7.3	Trusted IELSR Setting Procedure .....	358
13.8	Return from Low Power Modes .....	359
13.8.1	Return from CPU Sleep Mode .....	359
13.8.2	Return from CPU Deep Sleep Mode.....	359
13.8.3	Return from Software Standby Mode.....	360
13.9	Using the WFI Instruction with Non-Maskable Interrupts .....	360
13.10	Reference .....	360
<b>14.</b>	<b>Buses.....</b>	<b>361</b>
14.1	Overview.....	361
14.2	Description of Buses.....	363
14.2.1	Arbitration .....	363
14.2.2	External Bus .....	364
14.2.3	Parallel Operation .....	364
14.2.4	Restrictions .....	364
14.3	Register Descriptions .....	364
14.3.1	BUSSARA : Bus Security Attribution Register A.....	364
14.3.2	BUSSARB : Bus Security Attribution Register B.....	365

14.3.3	BUSDIVBYP : Bus Divider Bypass Register .....	366
14.3.4	BUSSABT0<slave> : Bus Slave Arbitration Control Register 0(<slave> = FLBI, STBYSBI, EOBI, PBBI, PABI, PIBI, PSBI) .....	367
14.3.5	BUSSABT1<slave> : Bus Slave Arbitration Control Register 1(<slave> = FHBI, S1BI) .....	368
14.3.6	BUSnERRADD : BUS Error Address Register (n = 4, 5) .....	369
14.3.7	BUSnERRRW : BUS Error Read Write (n = 4, 5) .....	370
14.3.8	BMSAnERRADD : Bus Master Security Attribution Unit Error Address (n = 4, 5) .....	370
14.3.9	BMSAnERRRW : BUS Master Security Attribution Unit Error Read Write (n = 4, 5) .....	371
14.3.10	BUSnERRSTAT : BUS Error Status Register (n = 1 to 5, 10) .....	371
14.3.11	BUSnERRCLR : BUS Error Clear Register (n = 1 to 5, 10) .....	373
14.3.12	BUSOAD : BUS Operation After Detection Register .....	374
14.3.13	BUSOADPT : BUS Operation After Detection Protect Register .....	375
14.3.14	MBWERRSTAT : Master Bufferable Write Error Status Register .....	375
14.3.15	MBWERRCLR : Master Bufferable Write Error Clear Register .....	376
14.3.16	SBWERRSTAT : Slave Bufferable Write Error Status Register .....	377
14.3.17	SBWERRCLR : Slave Bufferable Write Error Clear Register .....	379
14.4	Bus Error Monitoring Section .....	379
14.4.1	Bus Error Types .....	380
14.4.2	Operations When a Bus Error Occurs .....	380
14.4.3	Conditions Leading to Illegal Address Access Errors .....	382
14.4.4	Timeout .....	384
14.5	References .....	384
<b>15.</b>	<b>Memory Protection Unit (MPU) .....</b>	<b>386</b>
15.1	Overview .....	386
15.2	Arm MPU .....	386
15.3	Bus master MPU .....	386
15.3.1	Register Descriptions .....	387
15.3.2	Operation .....	401
15.4	References .....	405
<b>16.</b>	<b>DMA Controller (DMAC) .....</b>	<b>406</b>
16.1	Overview .....	406
16.2	Register Descriptions .....	408
16.2.1	DMACCHSAR : DMA channel Security Attribution Register .....	408
16.2.2	DMACCHPAR : DMA channel Privilege Attribution Register .....	409
16.2.3	DMACCSAR : DMAC Controller Security Attribution Register .....	410
16.2.4	DELSRn : DMAC Event Link Setting Register n (n = 0 to 7) .....	410
16.2.5	DMSAR : DMA Source Address Register .....	412
16.2.6	DMSRR : DMA Source Reload Address Register .....	412
16.2.7	DMDAR : DMA Destination Address Register .....	412
16.2.8	DMDRR : DMA Destination Reload Address Register .....	413

16.2.9	DMCRA : DMA Transfer Count Register .....	413
16.2.10	DMCRB : DMA Block Transfer Count Register .....	414
16.2.11	DMTMD : DMA Transfer Mode Register .....	415
16.2.12	DMINT : DMA Interrupt Setting Register .....	416
16.2.13	DMAMD : DMA Address Mode Register .....	417
16.2.14	DMOFR : DMA Offset Register .....	420
16.2.15	DMCNT : DMA Transfer Enable Register .....	420
16.2.16	DMREQ : DMA Software Start Register .....	421
16.2.17	DMSTS : DMA Status Register .....	422
16.2.18	DMSBS : DMA Source Buffer Size Register .....	423
16.2.19	DMDBS : DMA Destination Buffer Size Register .....	424
16.2.20	DMBWR : DMA Bufferable Write Enable Register .....	425
16.2.21	DMAST : DMA Module Activation Register .....	426
16.2.22	DMECHR : DMAC Error Channel Register .....	426
16.3	Operation .....	428
16.3.1	Transfer Mode .....	428
16.3.2	Extended Repeat Area Function .....	437
16.3.3	Free-running Function .....	439
16.3.4	Address Update Function using Offset .....	440
16.3.5	Address Update Function in Repeat-Block Transfer Mode .....	445
16.3.6	Example of Using Repeat-Block Transfer Mode .....	447
16.3.7	Activation Sources .....	450
16.3.8	Operation Timing .....	450
16.3.9	Activating the DMAC .....	451
16.3.10	Starting DMA Transfer .....	453
16.3.11	Registers during DMA Transfer .....	453
16.3.12	Channel Priority .....	455
16.3.13	Channel Security .....	455
16.3.14	Channel Privilege .....	455
16.4	Ending DMA Transfer .....	455
16.4.1	Transfer End by Completion of Specified Total Number of Transfer Operations .....	456
16.4.2	Transfer End by Repeat Size End Interrupt .....	456
16.4.3	Transfer End by Interrupt on Extended Repeat Area Overflow .....	456
16.5	Processing on DMA Transfer Error .....	457
16.6	Interrupts .....	457
16.6.1	Transfer End Interrupt .....	457
16.6.2	Transfer Error Interrupt .....	459
16.7	Event Link .....	460
16.8	Low-Power Consumption Function .....	460
16.9	Usage Notes .....	460

16.9.1	Access to the Registers during DMA Transfer .....	460
16.9.2	DMA Transfer to Reserved Areas .....	461
16.9.3	Setting of DMAC Event Link Setting Register of the Interrupt Controller Unit (DELSRn) ...	461
16.9.4	Suspending or Restarting DMAC Activation .....	461
16.9.5	Precautions for Resuming DMA Transfer.....	461
16.9.6	Restriction for OSPI .....	462
<b>17.</b>	<b>Data Transfer Controller (DTC).....</b>	<b>463</b>
17.1	Overview.....	463
17.2	Register Descriptions .....	464
17.2.1	DTCSAR : DTC Controller Security Attribution Register .....	464
17.2.2	MRA : DTC Mode Register A .....	465
17.2.3	MRB : DTC Mode Register B .....	466
17.2.4	SAR : DTC Transfer Source Register .....	467
17.2.5	DAR : DTC Transfer Destination Register.....	467
17.2.6	CRA : DTC Transfer Count Register A.....	467
17.2.7	CRB : DTC Transfer Count Register B.....	468
17.2.8	DTCCR : DTC Control Register .....	469
17.2.9	DTCCR_SEC : DTC Control Register for Secure Region .....	469
17.2.10	DTCVBR : DTC Vector Base Register .....	470
17.2.11	DTCVBR_SEC : DTC Vector Base Register for Secure Region .....	470
17.2.12	DTCST : DTC Module Start Register .....	470
17.2.13	DTCSTS : DTC Status Register .....	471
17.2.14	DTEVR : DTC Error Vector Register .....	472
17.3	Activation Sources.....	473
17.3.1	Allocating Transfer Information and DTC Vector Table .....	473
17.4	Operation.....	476
17.4.1	Transfer Information Read Skip Function.....	478
17.4.2	Transfer Information Write-Back Skip Function.....	478
17.4.3	Normal Transfer Mode .....	479
17.4.4	Repeat Transfer Mode .....	480
17.4.5	Block Transfer Mode .....	481
17.4.6	Chain Transfer.....	482
17.4.7	Operation Timing.....	483
17.4.8	Vector Security .....	486
17.4.9	Vector privilege.....	486
17.5	DTC Setting Procedure .....	486
17.6	Examples of DTC Usage .....	487
17.6.1	Normal Transfer .....	487
17.6.2	Chain transfer.....	487
17.6.3	Chain Transfer when Counter = 0 .....	489

17.7	Processing on DTC Transfer Error .....	490
17.8	Interrupt .....	491
17.8.1	Interrupt Request of Transfer End .....	491
17.8.2	Interrupt Request of Transfer Error .....	491
17.9	Event Link .....	491
17.10	Low Power Consumption Function .....	491
17.11	Usage Notes .....	492
17.11.1	Transfer Information Start Address .....	492
17.11.2	Restriction for OSPI .....	492
<b>18.</b>	<b>Event Link Controller (ELC).....</b>	<b>493</b>
18.1	Overview.....	493
18.2	Register Descriptions .....	495
18.2.1	ELCR : Event Link Controller Register .....	495
18.2.2	ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1).....	495
18.2.3	ELSRn : Event Link Setting Register n (n = 0 to 12, 14 to 17).....	496
18.2.4	ELCSARA : Event Link Controller Security Attribution Register A .....	502
18.2.5	ELCSARB : Event Link Controller Security Attribution Register B .....	503
18.2.6	ELCPARA : Event Link Controller Privilege Attribution Register A.....	503
18.2.7	ELCPARB : Event Link Controller Privilege Attribution Register B.....	504
18.3	Operation.....	504
18.3.1	Relation between Interrupt Handling and Event Linking .....	504
18.3.2	Linking Events .....	504
18.3.3	Example of Procedure for Linking Events .....	505
18.4	Usage Notes.....	505
18.4.1	Linking DMAC/DTC Transfer End Signals as Events.....	505
18.4.2	Setting Clocks .....	505
18.4.3	Module-Stop Function Setting.....	505
18.4.4	ELC Delay Time .....	506
<b>19.</b>	<b>I/O Ports.....</b>	<b>507</b>
19.1	Overview.....	507
19.2	Register Descriptions .....	509
19.2.1	PCNTR1/PODR/PDR : Port Control Register 1 .....	509
19.2.2	PCNTR2/EIDR/PIDR : Port Control Register 2 .....	510
19.2.3	PCNTR3/PORR/POSR : Port Control Register 3.....	511
19.2.4	PCNTR4/EORR/EOSR : Port Control Register 4.....	512
19.2.5	PmnPFS/PmnPFS_HA/PmnPFS_BY : Port mn Pin Function Select Register (m = 0 to 9, n = 00 to 15).....	513
19.2.6	PWPR_NS : Write-Protect Register for Non-secure .....	515
19.2.7	PFENET : Ethernet Control Register .....	515
19.2.8	PWPR_S : Write-Protect Register for Secure .....	516

19.2.9	PmSAR : Port Security Attribution register (m = 0 to 9) .....	516
19.2.10	LVOCR : Low Voltage Operation Control register .....	517
19.3	Operation .....	517
19.3.1	General I/O Ports .....	517
19.3.2	Port Function Select .....	517
19.3.3	Port Group Function for ELC .....	518
19.4	Handling of Unused Pins .....	520
19.5	Usage Notes .....	521
19.5.1	Procedure for Specifying the Pin Functions .....	521
19.5.2	Procedure for Using Port Group Input .....	521
19.5.3	Port Output Data Register (PODR) Summary .....	521
19.5.4	Notes on Using Analog Functions .....	521
19.5.5	I/O Buffer Specification .....	521
19.5.6	Selecting Pins USB_DP and USB_DM .....	523
19.5.7	Pull-up/Pull-down Setting for P814 and P815 at USBFS/GPIO Function .....	524
19.6	Peripheral Select Settings for Each Product .....	524
<b>20.</b>	<b>Port Output Enable for GPT (POEG) .....</b>	<b>532</b>
20.1	Overview .....	532
20.2	Register Descriptions .....	533
20.2.1	POEGGn : POEG Group n Setting Register (n = A to D) .....	533
20.3	Output-Disable Control Operation .....	534
20.3.1	Pin Input Level Detection Operation .....	535
20.3.2	Output-Disable Requests from the GPT .....	536
20.3.3	Comparator Interrupt Detection .....	536
20.3.4	Output-Disable Control Using Detection of Stopped Oscillation .....	536
20.3.5	Output-Disable Control Using Registers .....	536
20.3.6	Release from Output-Disable .....	536
20.4	Interrupt Sources .....	537
20.5	External Trigger Output to the GPT .....	537
20.6	Usage Notes .....	538
20.6.1	Transition to Software Standby Mode .....	538
20.6.2	Specifying Pins Associated with the GPT .....	538
<b>21.</b>	<b>General PWM Timer (GPT) .....</b>	<b>539</b>
21.1	Overview .....	539
21.2	Register Descriptions .....	542
21.2.1	GTWP : General PWM Timer Write-Protection Register .....	542
21.2.2	GTSTR : General PWM Timer Software Start Register .....	544
21.2.3	GTSTP : General PWM Timer Software Stop Register .....	545
21.2.4	GTCLR : General PWM Timer Software Clear Register .....	546

21.2.5	GTSSR : General PWM Timer Start Source Select Register .....	547
21.2.6	GTPSR : General PWM Timer Stop Source Select Register .....	550
21.2.7	GTCSR : General PWM Timer Clear Source Select Register .....	553
21.2.8	GTUPSR : General PWM Timer Up Count Source Select Register .....	556
21.2.9	GTDNSR : General PWM Timer Down Count Source Select Register .....	559
21.2.10	GTICASR : General PWM Timer Input Capture Source Select Register A.....	563
21.2.11	GTICBSR : General PWM Timer Input Capture Source Select Register B.....	566
21.2.12	GTCR : General PWM Timer Control Register .....	569
21.2.13	GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register .....	571
21.2.14	GTIOR : General PWM Timer I/O Control Register .....	573
21.2.15	GTINTAD : General PWM Timer Interrupt Output Setting Register .....	577
21.2.16	GTST : General PWM Timer Status Register .....	579
21.2.17	GTBER : General PWM Timer Buffer Enable Register .....	584
21.2.18	GTCNT : General PWM Timer Counter .....	587
21.2.19	GTCCRk : General PWM Timer Compare Capture Register k (k = A to F) .....	587
21.2.20	GTPR : General PWM Timer Cycle Setting Register .....	588
21.2.21	GTPBR : General PWM Timer Cycle Setting Buffer Register .....	588
21.2.22	GTADTRk : A/D Conversion Start Request Timing Register k (k = A, B).....	588
21.2.23	GTADTBRk : A/D Conversion Start Request Timing Buffer Register k (k = A, B).....	589
21.2.24	GTADTDBRk : A/D Conversion Start Request Timing Double-Buffer Register k (k = A, B).....	589
21.2.25	GTDTCR : General PWM Timer Dead Time Control Register .....	590
21.2.26	GTDVU : General PWM Timer Dead Time Value Register U.....	590
21.2.27	GTADSMR : General PWM Timer A/D Conversion Start Request Signal Monitoring Register.....	591
21.2.28	GTICLF : General PWM Timer Inter Channel Logical Operation Function Setting Register .....	592
21.2.29	GTPC : General PWM Timer Period Count Register .....	594
21.2.30	GTSECSR : General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register .....	595
21.2.31	GTSECR : General PWM Timer Operation Enable Bit Simultaneous Control Register.....	596
21.3	Operation.....	598
21.3.1	Basic Operation.....	598
21.3.2	Buffer Operation .....	606
21.3.3	PWM Output Operating Mode.....	616
21.3.4	Automatic Dead Time Setting Function.....	626
21.3.5	Count Direction Changing Function .....	630
21.3.6	Function of Output Duty 0% and 100%.....	631
21.3.7	Hardware Count Start/Count Stop and Clear Operation .....	632
21.3.8	Synchronized Operation.....	637
21.3.9	PWM Output Operation Examples .....	642
21.3.10	Period Count Function .....	647



21.3.11	Phase Counting Function .....	648
21.3.12	Inter Channel Logical Operation Function.....	658
21.4	Interrupt Sources .....	660
21.4.1	Interrupt Sources.....	660
21.4.2	DMAC and DTC Activation.....	662
21.5	A/D Conversion Start Request .....	662
21.6	Operations Linked by ELC.....	665
21.6.1	Event Signal Output to ELC .....	665
21.6.2	Event Signal Inputs from ELC .....	665
21.7	Noise Filter Function.....	665
21.8	Protection Function.....	666
21.8.1	Write-Protection for Registers .....	666
21.8.2	Disabling of Buffer Operation .....	666
21.8.3	GTIOCNm Pin Output Negate Control (n = 0 to 5, 10 to 13, m = A, B) .....	668
21.9	Initialization Method of Output Pins .....	669
21.9.1	Pin Settings after Reset .....	669
21.9.2	Pin Initialization Due to Error during Operation.....	670
21.10	Usage Notes.....	670
21.10.1	Module-Stop Function Setting.....	670
21.10.2	GTCCRn Settings during Compare Match Operation (n = A to F).....	670
21.10.3	Setting Range for GTCNT Counter .....	671
21.10.4	Starting and Stopping the GTCNT Counter .....	671
21.10.5	Priority Order of Each Event .....	671
21.10.6	Prohibit Invalid Register Setting .....	672
<b>22.</b>	<b>Low Power Asynchronous General Purpose Timer (AGT).....</b>	<b>673</b>
22.1	Overview.....	673
22.2	Register Descriptions .....	675
22.2.1	AGT : AGT Counter Register .....	675
22.2.2	AGTCMA : AGT Compare Match A Register .....	675
22.2.3	AGTCMB : AGT Compare Match B Register .....	676
22.2.4	AGTCR : AGT Control Register .....	676
22.2.5	AGTMR1 : AGT Mode Register 1 .....	678
22.2.6	AGTMR2 : AGT Mode Register 2 .....	679
22.2.7	AGTIOC : AGT I/O Control Register .....	680
22.2.8	AGTISR : AGT Event Pin Select Register.....	681
22.2.9	AGTCMSR : AGT Compare Match Function Select Register .....	682
22.2.10	AGTIOSEL : AGT Pin Select Register .....	682
22.3	Operation.....	683
22.3.1	Reload Register and Counter Rewrite Operation.....	683
22.3.2	Reload Register and AGT Compare Match A/B Register Rewrite Operation .....	684

22.3.3	Timer Mode .....	685
22.3.4	Pulse Output Mode .....	686
22.3.5	Event Counter Mode .....	687
22.3.6	Pulse Width Measurement Mode .....	688
22.3.7	Pulse Period Measurement Mode .....	689
22.3.8	Compare Match function .....	690
22.3.9	Output Settings for Each Mode .....	691
22.3.10	Standby Mode .....	692
22.3.11	Interrupt Sources.....	693
22.3.12	Event Signal Output to ELC .....	693
22.4	Usage Notes.....	694
22.4.1	Count Operation Start and Stop Control .....	694
22.4.2	Access to Counter Register .....	694
22.4.3	When Changing Mode .....	694
22.4.4	Output pin setting .....	694
22.4.5	Digital Filter .....	694
22.4.6	How to Calculate Event Number, Pulse Width, and Pulse Period.....	694
22.4.7	When Count is Forcibly Stopped by TSTOP Bit.....	695
22.4.8	When Selecting AGT0 Underflow as the Count Source.....	695
22.4.9	Module-stop function.....	695
22.4.10	When Switching Source Clock .....	695
<b>23.</b>	<b>Ultra-Low-Power Timer (ULPT) .....</b>	<b>696</b>
23.1	Overview.....	696
23.2	Configuration .....	696
23.3	Register Descriptions .....	697
23.3.1	ULPTCNT : ULPT Counter Register .....	697
23.3.2	ULPTCMA : ULPT Compare Match A Register .....	698
23.3.3	ULPTCMB : ULPT Compare Match B Register .....	698
23.3.4	ULPTCR : ULPT Control Register.....	699
23.3.5	ULPTMR1 : ULPT Mode Register 1 .....	701
23.3.6	ULPTMR2 : ULPT Mode Register 2.....	701
23.3.7	ULPTMR3 : ULPT Mode Register 3.....	702
23.3.8	ULPTIOC : ULPT I/O Control Register.....	704
23.3.9	ULPTISR : ULPT Event Pin Select Register .....	704
23.3.10	ULPTCMSR : ULPT Compare Match Function Select Register .....	705
23.4	Operation.....	706
23.4.1	Count operation.....	706
23.4.2	Rewriting the counter and reload register .....	716
23.4.3	Rewriting the compare circuits and reload registers for compare match A/B .....	719
23.4.4	Pulse output .....	720

23.4.5	Compare match function .....	721
23.4.6	Input and output settings for each mode .....	722
23.4.7	Standby mode .....	723
23.4.8	Interrupt sources .....	724
23.4.9	Event output to the ELC .....	725
23.5	Usage notes .....	725
23.5.1	Start and stop control of the counter .....	725
23.5.2	Access to flags (TUNDF, TCMAF, and TCMBF bits in ULPTCR register).....	726
23.5.3	Access to the ULPTCNT, ULPTCMA, and ULPTCMB registers .....	726
23.5.4	Mode changes.....	726
23.5.5	Setting the ULPTOn, ULPTOAn, and ULPTOBn pins.....	726
23.5.6	Calculating the number of events.....	726
23.5.7	When the counter is stopped forcibly by the TSTOP bit.....	726
23.5.8	Digital filter .....	726
23.5.9	Restrictions on Event Counting during Deep Software Standby Mode 1.....	726
23.5.10	Module-stop function.....	727
23.5.11	Setting the ULPTEEn and ULPTEVIn pins .....	727
<b>24.</b>	<b>Realtime Clock (RTC).....</b>	<b>728</b>
24.1	Overview.....	728
24.2	Register Descriptions .....	729
24.2.1	R64CNT : 64-Hz Counter.....	730
24.2.2	RSECCNT : Second Counter (in Calendar Count Mode) .....	730
24.2.3	RMINCNT : Minute Counter (in Calendar Count Mode).....	731
24.2.4	RHRCNT : Hour Counter (in Calendar Count Mode).....	731
24.2.5	RWKCNT : Day-of-Week Counter (in Calendar Count Mode) .....	732
24.2.6	BCNTn : Binary Counter n (n = 0 to 3) (in Binary Count Mode).....	732
24.2.7	RDAYCNT : Day Counter.....	733
24.2.8	RMONCNT : Month Counter .....	733
24.2.9	RYRCNT : Year Counter .....	734
24.2.10	RSECAR : Second Alarm Register (in Calendar Count Mode).....	734
24.2.11	RMINAR : Minute Alarm Register (in Calendar Count Mode).....	735
24.2.12	RHRAR : Hour Alarm Register (in Calendar Count Mode) .....	735
24.2.13	RWKAR : Day-of-Week Alarm Register (in Calendar Count Mode).....	736
24.2.14	BCNTnAR : Binary Counter n Alarm Register (n = 0 to 3) (in Binary Count Mode) .....	737
24.2.15	RDAYAR : Date Alarm Register (in Calendar Count Mode).....	737
24.2.16	RMONAR : Month Alarm Register (in Calendar Count Mode).....	738
24.2.17	RYRAR : Year Alarm Register (in Calendar Count Mode).....	739
24.2.18	RYRAREN : Year Alarm Enable Register (in Calendar Count Mode).....	739
24.2.19	BCNTnAER : Binary Counter n Alarm Enable Register (n = 0 to 3) (in Binary Count Mode).....	740

24.2.20	RCR1 : RTC Control Register 1 .....	740
24.2.21	RCR2 : RTC Control Register 2 (in Calendar Count Mode).....	741
24.2.22	RCR2 : RTC Control Register 2 (in Binary Count Mode) .....	743
24.2.23	RCR4 : RTC Control Register 4 .....	745
24.2.24	RFRL : Frequency Register L .....	745
24.2.25	RFRH : Frequency Register H .....	746
24.2.26	RADJ : Time Error Adjustment Register.....	746
24.2.27	RTCCRN : Time Capture Control Register n (n = 0 to 2).....	747
24.2.28	RSECCPn : Second Capture Register n (n = 0 to 2) (in Calendar Count Mode).....	748
24.2.29	RMINCPn : Minute Capture Register n (n = 0 to 2) (in Calendar Count Mode).....	748
24.2.30	RHRCPn : Hour Capture Register n (n = 0 to 2) (in Calendar Count Mode).....	749
24.2.31	RDAYCPn : Date Capture Register n (n = 0 to 2) (in Calendar Count Mode).....	749
24.2.32	RMONCPn : Month Capture Register n (n = 0 to 2) (in Calendar Count Mode) .....	750
24.2.33	BCNTnCPm : BCNTn Capture Register m (n = 0 to 3, m = 0 to 2) (in Binary Count Mode) .....	750
24.3	Operation.....	751
24.3.1	Outline of Initial Settings of Registers after Power On .....	751
24.3.2	Clock and Count Mode Setting Procedure .....	751
24.3.3	Setting the Time .....	752
24.3.4	30-Second Adjustment.....	753
24.3.5	Reading 64-Hz Counter and Time.....	754
24.3.6	Alarm Function .....	756
24.3.7	Procedure for Disabling Alarm Interrupt.....	757
24.3.8	Time Error Adjustment Function.....	757
24.3.9	Capturing the time .....	759
24.4	Interrupt Sources .....	760
24.5	Event Link Output.....	761
24.5.1	Interrupt Handling and Event Linking .....	762
24.6	Usage Notes.....	762
24.6.1	Register Writing during Counting .....	762
24.6.2	Use of Periodic Interrupts.....	762
24.6.3	RTCOU (1-Hz/64-Hz) Clock Output.....	763
24.6.4	Transitions to Low Power Modes after Setting Registers.....	763
24.6.5	Notes on Writing to and Reading from Registers.....	763
24.6.6	Changing the Count Mode .....	763
24.6.7	Initialization Procedure When the RTC Is Not to Be Used.....	763
24.6.8	When Switching Source Clock .....	764
24.6.9	When writing to the RTC registers .....	764
<b>25.</b>	<b>Watchdog Timer (WDT).....</b>	<b>765</b>
25.1	Overview.....	765

25.2	Register Descriptions .....	766
25.2.1	WDTRR : WDT Refresh Register.....	766
25.2.2	WDTCR : WDT Control Register.....	767
25.2.3	WDTSR : WDT Status Register .....	769
25.2.4	WDTRCR : WDT Reset Control Register.....	770
25.2.5	WDCSTPR : WDT Count Stop Control Register.....	771
25.2.6	Option Function Select Register 0 (OFS0).....	771
25.3	Operation.....	771
25.3.1	Count Operation in each Start Mode.....	771
25.3.2	Controlling Writes to the WDTCR, WDTRCR, and WDCSTPR Registers.....	775
25.3.3	Refresh Operation.....	775
25.3.4	Status Flags .....	776
25.3.5	Reset Output.....	776
25.3.6	Interrupt Sources.....	776
25.3.7	Reading the Down-Counter Value.....	777
25.3.8	Association between Option Function Select Register 0 (OFS0) and WDT Registers .....	777
25.4	Output to the Event Link Controller (ELC).....	778
25.5	Usage Notes.....	778
25.5.1	ICU Event Link Setting Register n (IELSRn) Setting.....	778
<b>26.</b>	<b>Independent Watchdog Timer (IWDT).....</b>	<b>779</b>
26.1	Overview.....	779
26.1.1	Block diagram .....	779
26.2	Register specifications.....	780
26.2.1	IWDTRR : IWDT Refresh Register.....	780
26.2.2	IWDTCR : IWDT Control Register.....	781
26.2.3	IWDTSR : IWDT Status Register .....	783
26.2.4	IWDTRCR : IWDT Reset Control Register.....	784
26.2.5	IWDCSTPR : IWDT Count Stop Control Register.....	785
26.3	Operation.....	785
26.3.1	Count operation in each start mode.....	785
26.3.2	Controlling writes to the IWDTCR, IWDTRCR, and IWDCSTPR registers .....	788
26.3.3	Refresh operation.....	788
26.3.4	Status flags .....	791
26.3.5	Reset output.....	791
26.3.6	Interrupt sources .....	791
26.3.7	Reading the down-counter value .....	791
26.3.8	Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers .....	793
26.4	Link operation by the event link function .....	793
<b>27.</b>	<b>Ethernet MAC Controller (ETHERC) .....</b>	<b>794</b>

27.1	Overview.....	794
27.2	Register Descriptions .....	799
27.2.1	ECMR : ETHERC Mode Register .....	799
27.2.2	RFLR : Receive Frame Maximum Length Register .....	801
27.2.3	ECSR : ETHERC Status Register.....	801
27.2.4	ECSIPR : ETHERC Interrupt Enable Register .....	802
27.2.5	PIR : PHY Interface Register .....	803
27.2.6	PSR : PHY Status Register.....	804
27.2.7	RDMLR : Random Number Generation Counter Upper Limit Setting Register .....	804
27.2.8	IPGR : Interpacket Gap Register .....	804
27.2.9	APR : Automatic PAUSE Frame Register .....	805
27.2.10	MPR : Manual PAUSE Frame Register.....	805
27.2.11	RFCF : Received PAUSE Frame Counter.....	806
27.2.12	TPAUSER : PAUSE Frame Retransmit Count Setting Register.....	806
27.2.13	TPAUSECR : PAUSE Frame Retransmit Counter .....	807
27.2.14	BCFRR : Broadcast Frame Receive Count Setting Register .....	807
27.2.15	MAHR : MAC Address Upper Bit Register .....	807
27.2.16	MALR : MAC Address Lower Bit Register.....	808
27.2.17	TROCR : Transmit Retry Over Counter Register .....	808
27.2.18	CDCR : Late Collision Detect Counter Register.....	809
27.2.19	LCCR : Lost Carrier Counter Register .....	809
27.2.20	CNDCR : Carrier Not Detect Counter Register .....	809
27.2.21	CEFCR : CRC Error Frame Receive Counter Register .....	810
27.2.22	FRECR : Frame Receive Error Counter Register .....	810
27.2.23	TSFRCR : Too-Short Frame Receive Counter Register .....	810
27.2.24	TLFRCR : Too-Long Frame Receive Counter Register .....	811
27.2.25	RFCR : Received Alignment Error Frame Counter Register.....	811
27.2.26	MAFCR : Multicast Address Frame Receive Counter Register .....	812
27.3	Operation.....	812
27.3.1	Transmission .....	812
27.3.2	Reception .....	814
27.3.3	Frame Timing .....	815
27.3.4	Accessing the MII and RMII Registers .....	817
27.3.5	Magic Packet Detection .....	819
27.3.6	Adjusting Transmission Efficiency by Changing the IPG .....	819
27.3.7	Flow Control .....	820
27.4	Interrupts .....	822
27.5	Usage Notes.....	822
27.5.1	Preventing the LCHNG Flag from Erroneously Setting to 1 .....	822
27.5.2	Input to RMII0_RX_ER Pin While RMII Is Selected .....	822

27.5.3	Collision Occurrence in Half-Duplex Mode .....	822
<b>28.</b>	<b>Ethernet DMA Controller (EDMAC) .....</b>	<b>823</b>
28.1	Overview.....	823
28.2	Register Descriptions .....	824
28.2.1	EDMR : EDMAC Mode Register .....	824
28.2.2	EDTRR : EDMAC Transmit Request Register .....	825
28.2.3	EDRRR : EDMAC Receive Request Register.....	826
28.2.4	TDLAR : Transmit Descriptor List Start Address Register .....	826
28.2.5	RDLAR : Receive Descriptor List Start Address Register.....	827
28.2.6	EESR : ETHERC/EDMAC Status Register .....	827
28.2.7	EESIPR : ETHERC/EDMAC Status Interrupt Enable Register .....	830
28.2.8	TRSCER : ETHERC/EDMAC Transmit/Receive Status Copy Enable Register.....	832
28.2.9	RMFCR : Missed-Frame Counter Register .....	833
28.2.10	TFTR : Transmit FIFO Threshold Register.....	833
28.2.11	FDR : FIFO Depth Register.....	834
28.2.12	RMCR : Receive Method Control Register .....	834
28.2.13	TFUCR : Transmit FIFO Underflow Counter .....	835
28.2.14	RFOCR : Receive FIFO Overflow Counter .....	835
28.2.15	IOSR : Independent Output Signal Setting Register .....	836
28.2.16	FCFTR : Flow Control Start FIFO Threshold Setting Register .....	836
28.2.17	RPADIR : Receive Data Padding Insert Register.....	837
28.2.18	TRIMD : Transmit Interrupt Setting Register .....	837
28.2.19	RBWAR : Receive Buffer Write Address Register.....	838
28.2.20	RDFAR : Receive Descriptor Fetch Address Register .....	838
28.2.21	TBRAR : Transmit Buffer Read Address Register .....	839
28.2.22	TDFAR : Transmit Descriptor Fetch Address Register.....	839
28.3	Operation.....	839
28.3.1	Descriptor Lists and Data Buffers.....	839
28.3.2	Transmission .....	843
28.3.3	Reception .....	844
28.3.4	Multi-Buffer Frame Transmission .....	845
28.3.5	Bus Transfer Error.....	847
28.4	Interrupts .....	849
28.5	Usage Notes.....	849
28.5.1	Settings for the Module-Stop Function.....	849
28.5.2	Stopping the EDMAC during Operation .....	850
<b>29.</b>	<b>USB 2.0 Full-Speed Module (USBFS) .....</b>	<b>851</b>
29.1	Overview.....	851
29.2	Register Descriptions .....	853

29.2.1	SYSCFG : System Configuration Control Register .....	853
29.2.2	SYSSTS0 : System Configuration Status Register 0 .....	854
29.2.3	DVSTCTR0 : Device State Control Register 0 .....	855
29.2.4	CFIFO/CFIFOL : CFIFO Port Register .....	858
29.2.5	DnFIFO/DnFIFOL : DnFIFO Port Register (n = 0, 1) .....	859
29.2.6	CFIFOSEL : CFIFO Port Select Register .....	860
29.2.7	DnFIFOSEL : DnFIFO Port Select Register (n = 0, 1) .....	862
29.2.8	CFIFOCTR : CFIFO Port Control Register .....	863
29.2.9	DnFIFOCTR : DnFIFO Port Control Register (n = 0, 1) .....	865
29.2.10	INTENB0 : Interrupt Enable Register 0 .....	866
29.2.11	INTENB1 : Interrupt Enable Register 1 .....	867
29.2.12	BRDYENB : BRDY Interrupt Enable Register .....	868
29.2.13	NRDYENB : NRDY Interrupt Enable Register .....	869
29.2.14	BEMPENB : BEMP Interrupt Enable Register .....	870
29.2.15	SOFCFG : SOF Output Configuration Register .....	871
29.2.16	INTSTS0 : Interrupt Status Register 0 .....	872
29.2.17	INTSTS1 : Interrupt Status Register 1 .....	874
29.2.18	BRDYSTS : BRDY Interrupt Status Register .....	877
29.2.19	NRDYSTS : NRDY Interrupt Status Register .....	878
29.2.20	BEMPSTS : BEMP Interrupt Status Register .....	878
29.2.21	FRMNUM : Frame Number Register .....	879
29.2.22	DVCHGR : Device State Change Register .....	880
29.2.23	USBADDR : USB Address Register .....	880
29.2.24	USBREQ : USB Request Type Register .....	881
29.2.25	USBVAL : USB Request Value Register .....	882
29.2.26	USBINDX : USB Request Index Register .....	883
29.2.27	USBLENG : USB Request Length Register .....	883
29.2.28	DCPCFG : DCP Configuration Register .....	884
29.2.29	DCPMAXP : DCP Maximum Packet Size Register .....	885
29.2.30	DCPCTR : DCP Control Register .....	885
29.2.31	PIPESEL : Pipe Window Select Register .....	888
29.2.32	PIPECFG : Pipe Configuration Register .....	889
29.2.33	PEMPMAXP : Pipe Maximum Packet Size Register .....	891
29.2.34	PIPEPERI : Pipe Cycle Control Register .....	892
29.2.35	PIPEnCTR : PIPEn Control Registers (n = 1 to 5) .....	893
29.2.36	PIPEnCTR : PIPEn Control Registers (n = 6 to 9) .....	897
29.2.37	PIPEnTRE : PIPEn Transaction Counter Enable Register (n = 1 to 5) .....	899
29.2.38	PIPEnTRN : PIPEn Transaction Counter Register (n = 1 to 5) .....	900
29.2.39	DEVADDn : Device Address n Configuration Register (n = 0 to 5) .....	901
29.2.40	DPUSR0R : Deep Software Standby USB Transceiver Control/Pin Monitor Register .....	901



29.2.41	DPUSR1R : Deep Software Standby USB Suspend/Resume Interrupt Register .....	903
29.3	Operation.....	904
29.3.1	System Control.....	904
29.3.2	Interrupts.....	913
29.3.3	Interrupt Descriptions .....	917
29.3.4	Pipe Control .....	926
29.3.5	FIFO Buffer.....	930
29.3.6	FIFO Buffer Clearing .....	930
29.3.7	FIFO Port Functions.....	931
29.3.8	DMA Transfers (D0FIFO and D1FIFO Ports).....	932
29.3.9	Control Transfers Using the DCP .....	932
29.3.10	Bulk Transfers (Pipes 1 to 5).....	934
29.3.11	Interrupt Transfers (Pipes 6 to 9) .....	934
29.3.12	Isochronous Transfers (Pipes 1 and 2) .....	935
29.3.13	SOF Interpolation Function .....	942
29.3.14	Pipe Schedule .....	942
29.4	Usage Notes.....	943
29.4.1	Settings for the Module-Stop State .....	943
29.4.2	Clearing the Interrupt Status Register on Canceling Software Standby Mode .....	944
29.4.3	Clearing the Interrupt Status Register after Setting Up the Port Function .....	944
29.4.4	Setting Up the Port Function before setting USB function .....	944
<b>30.</b>	<b>Serial Communications Interface (SCI) .....</b>	<b>945</b>
30.1	Overview.....	945
30.2	Register Descriptions .....	950
30.2.1	RSR : Receive Shift Register .....	950
30.2.2	RDR/RDR_BY : Receive Data Register.....	951
30.2.3	TDR/TDRLH/TDRLH : Transmit Data Register.....	952
30.2.4	TSR : Transmit Shift Register.....	953
30.2.5	CCR0 : Common Control Register 0.....	953
30.2.6	CCR1 : Common Control Register 1.....	956
30.2.7	CCR2 : Common Control Register 2.....	960
30.2.8	CCR3 : Common Control Register 3.....	974
30.2.9	CCR4 : Common Control Register 4.....	980
30.2.10	ICR : Simple IIC Control Register .....	982
30.2.11	FCR : FIFO Control Register.....	984
30.2.12	MCR : Manchester Control Register .....	986
30.2.13	DCR : Driver Control Register.....	989
30.2.14	XCR0 : Simple LIN Control Register 0 .....	990
30.2.15	XCR1 : Simple LIN Control Register 1 .....	992
30.2.16	XCR2 : Simple LIN Control Register 2 .....	994

30.2.17	CSR : Common Status Register .....	995
30.2.18	ISR : Simple IIC Status Register .....	1000
30.2.19	FRSR : FIFO Receive Status Register.....	1001
30.2.20	FTSR : FIFO Transmit Status Register .....	1002
30.2.21	MSR : Manchester Status Register.....	1003
30.2.22	XSR0 : Simple LIN Status Register 0.....	1005
30.2.23	XSR1 : Simple LIN Status Register 1.....	1007
30.2.24	CFCLR : Common Flag Clear Register.....	1008
30.2.25	ICFCLR : Simple IIC Flag Clear Register.....	1009
30.2.26	FFCLR : FIFO Flag Clear Register .....	1009
30.2.27	MFCLR : Manchester Flag Clear Register .....	1010
30.2.28	XFCLR : Simple LIN Flag Clear Register .....	1010
30.2.29	CESR : Communication Enable Status Register .....	1011
30.3	Operation in Asynchronous Mode .....	1011
30.3.1	Serial Data Transfer Format.....	1012
30.3.2	Receive Data Sampling Timing and Reception Margin in Asynchronous Mode .....	1013
30.3.3	Clock .....	1014
30.3.4	Double-Speed Operation and Frequency of 6 or 4 Times the Bit Rate.....	1014
30.3.5	CTS and RTS Functions .....	1015
30.3.6	Address Match (Receive Data Match Detection) Function .....	1015
30.3.7	SCI Initialization in Asynchronous Mode.....	1018
30.3.8	Serial Data Transmission in Asynchronous Mode.....	1020
30.3.9	Serial Data Reception in Asynchronous Mode.....	1026
30.3.10	The function of adjust receive sampling timing (Asynchronous Mode).....	1033
30.3.11	The function of adjust transmit timing (Asynchronous Mode) .....	1036
30.4	Multi-Processor Communication Function.....	1041
30.4.1	Multi-Processor Serial Data Transmission .....	1042
30.4.2	Multi-Processor Serial Data Reception .....	1045
30.5	Operation in Manchester mode .....	1051
30.5.1	Frame Format .....	1052
30.5.2	Clock .....	1056
30.5.3	Initialization of the SCI in Manchester Mode.....	1056
30.5.4	Double-speed operation.....	1057
30.5.5	CTS and RTS functions.....	1057
30.5.6	Serial data transmission in Manchester mode .....	1058
30.5.7	Serial Data Reception in Manchester Mode.....	1061
30.5.8	Operation When Multi-Processor Bit Is Used.....	1065
30.5.9	Receive Retiming .....	1065
30.5.10	Polarity Setting for Manchester Code .....	1066
30.5.11	Errors in Manchester Mode.....	1067

30.6	Operation in Clock Synchronous Mode .....	1072
30.6.1	Clock .....	1072
30.6.2	CTS and RTS Functions .....	1072
30.6.3	SCI Initialization in Clock Synchronous Mode.....	1073
30.6.4	Serial Data Transmission in Clock Synchronous Mode .....	1074
30.6.5	Serial Data Reception in Clock Synchronous Mode .....	1078
30.6.6	Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode .....	1082
30.6.7	Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used .....	1084
30.7	Operation in Smart Card Interface Mode.....	1085
30.7.1	Example Connection .....	1085
30.7.2	Data Format (Except in Block Transfer Mode).....	1086
30.7.3	Block Transfer Mode .....	1087
30.7.4	Receive Data Sampling Timing and Reception Margin.....	1087
30.7.5	SCI Initialization (Smart Card Interface Mode).....	1088
30.7.6	Serial Data Transmission (Except in Block Transfer Mode).....	1089
30.7.7	Serial Data Reception (Except in Block Transfer Mode).....	1092
30.7.8	Clock Output Control.....	1094
30.8	Operation in Simple IIC Mode .....	1094
30.8.1	Generation of Start, Restart, and Stop Conditions.....	1096
30.8.2	Clock Synchronization.....	1097
30.8.3	SDAn Output Delay.....	1097
30.8.4	SCI Initialization in Simple IIC Mode .....	1098
30.8.5	Operation in Master Transmission in Simple IIC Mode.....	1099
30.8.6	Master Reception in Simple IIC Mode.....	1102
30.9	Operation in Simple SPI Mode .....	1104
30.9.1	States of Pins in Master and Slave Modes .....	1105
30.9.2	SS Function in Master Mode.....	1106
30.9.3	SS Function in Slave Mode.....	1106
30.9.4	Relationship between Clock and Transmit/Receive Data .....	1106
30.9.5	SCI Initialization in Simple SPI Mode.....	1107
30.9.6	Transmission and Reception of Serial Data in Simple SPI Mode .....	1107
30.9.7	Reception Sampling Timing Adjustment Function in Simple SPI Mode with internal clock used .....	1108
30.10	Bit Rate Modulation Function .....	1108
30.11	Simple LIN mode .....	1108
30.11.1	Simple LIN Start Frame Transmission .....	1109
30.11.2	Simple LIN Start Frame Reception .....	1111
30.11.3	Simple LIN Bus Conflict Detection Function .....	1116
30.11.4	Simple LIN Bit Rate Measurement Function .....	1117
30.12	Interrupt Sources.....	1118

30.12.1	Buffer Operation for SCIn_TXI and SCIn_RXI Interrupts .....	1118
30.12.2	Interrupts in Asynchronous, Manchester, Clock Synchronous, and Simple SPI Modes ....	1119
30.12.3	Interrupts in Smart Card Interface Mode .....	1120
30.12.4	Interrupts in Simple IIC Mode.....	1121
30.12.5	Interrupts in Simple LIN mode.....	1121
30.13	Event Linking .....	1122
30.14	Noise Cancellation Function.....	1124
30.15	RS-485 Driver Control Function .....	1125
30.16	Loopback Function .....	1125
30.17	Half-Duplex communication Function.....	1126
30.18	Synchronizer Bypass Function .....	1126
30.19	Usage Notes.....	1127
30.19.1	Settings for the Module-Stop Function.....	1127
30.19.2	SCI Operation during Low Power State .....	1127
30.19.3	Break Detection and Processing.....	1130
30.19.4	Mark State and Production of Breaks .....	1131
30.19.5	Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode).....	1131
30.19.6	Writing Data to TDR .....	1131
30.19.7	Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode).....	1131
30.19.8	Restrictions on Using DMAC or DTC .....	1132
30.19.9	Notes on Starting Transfer .....	1132
30.19.10	Limitations on Simple SPI Mode .....	1133
30.19.11	Notes on Transmit Enable bit (CCR0.TE) .....	1133
30.19.12	Notes on Simple LIN mode .....	1134
30.19.13	Notes on RS-485 Driver Control function.....	1134
30.19.14	Notes on Loopback function.....	1134
30.19.15	Notes regarding register access when operation clock (TCLK) is slower than bus clock (PCLK) .....	1134
30.19.16	Notes on interrupting operation.....	1134
30.19.17	Notes on CCR3.BPEN bit setting.....	1134
<b>31.</b>	<b>I<sup>2</sup>C Bus Interface (IIC).....</b>	<b>1135</b>
31.1	Overview.....	1135
31.2	Register Descriptions .....	1137
31.2.1	ICCR1 : I <sup>2</sup> C Bus Control Register 1 .....	1137
31.2.2	ICCR2 : I <sup>2</sup> C Bus Control Register 2 .....	1139
31.2.3	ICMR1 : I <sup>2</sup> C Bus Mode Register 1 .....	1142
31.2.4	ICMR2 : I <sup>2</sup> C Bus Mode Register 2 .....	1143
31.2.5	ICMR3 : I <sup>2</sup> C Bus Mode Register 3 .....	1144

31.2.6	ICFER : I <sup>2</sup> C Bus Function Enable Register .....	1146
31.2.7	ICSER : I <sup>2</sup> C Bus Status Enable Register .....	1148
31.2.8	ICIER : I <sup>2</sup> C Bus Interrupt Enable Register .....	1149
31.2.9	ICSR1 : I <sup>2</sup> C Bus Status Register 1 .....	1150
31.2.10	ICSR2 : I <sup>2</sup> C Bus Status Register 2 .....	1152
31.2.11	ICWUR : I <sup>2</sup> C Bus Wakeup Unit Register .....	1156
31.2.12	ICWUR2 : I <sup>2</sup> C Bus Wakeup Unit Register 2 .....	1157
31.2.13	SARLy : Slave Address Register Ly (y = 0 to 2) .....	1158
31.2.14	SARUy : Slave Address Register Uy (y = 0 to 2) .....	1159
31.2.15	ICBRL : I <sup>2</sup> C Bus Bit Rate Low-Level Register .....	1159
31.2.16	ICBRH : I <sup>2</sup> C Bus Bit Rate High-Level Register .....	1160
31.2.17	ICDRT : I <sup>2</sup> C Bus Transmit Data Register .....	1161
31.2.18	ICDRR : I <sup>2</sup> C Bus Receive Data Register .....	1161
31.2.19	ICDRS : I <sup>2</sup> C Bus Shift Register .....	1162
31.3	Operation .....	1162
31.3.1	Communication Data Format .....	1162
31.3.2	Initial Settings .....	1163
31.3.3	Master Transmit Operation .....	1164
31.3.4	Master Receive Operation .....	1168
31.3.5	Slave Transmit Operation .....	1173
31.3.6	Slave Receive Operation .....	1176
31.4	SCL Synchronization Circuit .....	1178
31.5	SDA Output Delay Function .....	1179
31.6	Digital Noise Filter Circuits .....	1179
31.7	Address Match Detection .....	1180
31.7.1	Slave-Address Match Detection .....	1180
31.7.2	Detection of General Call Address .....	1183
31.7.3	Device-ID Address Detection .....	1184
31.7.4	Host Address Detection .....	1185
31.8	Wakeup Function .....	1186
31.8.1	Normal Wakeup Mode 1 .....	1187
31.8.2	Normal Wakeup Mode 2 .....	1190
31.8.3	Command Recovery Mode and EEP Response Mode (Special Wakeup Modes) .....	1192
31.9	Automatic Low-Hold Function for SCL .....	1195
31.9.1	Function to Prevent Wrong Transmission of Transmit Data .....	1195
31.9.2	NACK Reception Transfer Suspension Function .....	1196
31.9.3	Function to Prevent Failure to Receive Data .....	1197
31.10	Arbitration-Lost Detection Functions .....	1198
31.10.1	Master Arbitration-Lost Detection (MALE Bit) .....	1198

31.10.2	Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit).....	1200
31.10.3	Slave Arbitration-Lost Detection (SALE Bit).....	1201
31.11	Start, Restart, and Stop Condition Issuing Function.....	1202
31.11.1	Issuing a Start Condition .....	1202
31.11.2	Issuing a Restart Condition .....	1202
31.11.3	Issuing a Stop Condition .....	1204
31.12	Bus Hanging .....	1205
31.12.1	Timeout Function.....	1205
31.12.2	Extra SCL Clock Cycle Output Function .....	1206
31.12.3	IIC Reset and Internal Reset.....	1207
31.13	SMBus Operation .....	1207
31.13.1	SMBus Timeout Measurement.....	1207
31.13.2	Packet Error Code (PEC).....	1209
31.13.3	SMBus Host Notification Protocol (Notify ARP Master Command).....	1209
31.14	Interrupt Sources .....	1209
31.14.1	Buffer Operation for IICn_TXI and IICn_RXI Interrupts.....	1210
31.15	State of Registers When Issuing Each Condition.....	1210
31.16	Event Link Output.....	1211
31.16.1	Interrupt Handling and Event Linking .....	1211
31.17	Usage Notes.....	1211
31.17.1	Settings for the Module-Stop Function.....	1211
31.17.2	Notes on Starting Transfer .....	1211
<b>32.</b>	<b>CAN with Flexible Data-rate (CANFD).....</b>	<b>1213</b>
32.1	Overview.....	1213
32.1.1	CANFD Module .....	1213
32.1.2	Clock restriction.....	1215
32.2	Register Descriptions .....	1215
32.2.1	Register Table .....	1215
32.2.2	CFDC0NCFG : Nominal Bitrate Configuration Register.....	1216
32.2.3	CFDC0CTR : Control Register .....	1217
32.2.4	CFDC0STS : Status Register.....	1222
32.2.5	CFDC0ERFL : Error Flag Register.....	1224
32.2.6	CFDC0DCFG : Data Bitrate Configuration Register .....	1229
32.2.7	CFDC0FDCFG : CANFD Configuration Register.....	1231
32.2.8	CFDC0FDCTR : CANFD Control Register.....	1233
32.2.9	CFDC0FDSTS : CANFD Status Register .....	1234
32.2.10	CFDC0FDCRC : CANFD CRC Register .....	1236
32.2.11	CFDGCFG : Global Configuration Register .....	1237
32.2.12	CFDGCTR : Global Control Register .....	1239
32.2.13	CFDGSTS : Global Status Register .....	1241

32.2.14	CFDGERFL : Global Error Flag Register .....	1242
32.2.15	CFDGTINTSTS : Global TX Interrupt Status Register .....	1243
32.2.16	CFDGTSC : Global Timestamp Counter Register.....	1245
32.2.17	CFDGAFLCTR : Global Acceptance Filter List Entry Control Register .....	1245
32.2.18	CFDGAFLCFG : Global Acceptance Filter List Configuration Register .....	1246
32.2.19	CFDGAFLIDr : Global Acceptance Filter List ID Registers (r = 1 to 16) .....	1246
32.2.20	CFDGAFLMr : Global Acceptance Filter List Mask Registers (r = 1 to 16).....	1247
32.2.21	CFDGAFLP0r : Global Acceptance Filter List Pointer 0 Registers (r = 1 to 16).....	1249
32.2.22	CFDGAFLP1r : Global Acceptance Filter List Pointer 1 Registers (r = 1 to 16).....	1251
32.2.23	CFDRMNB : RX Message Buffer Number Register .....	1252
32.2.24	CFDRMND : RX Message Buffer New Data Register.....	1252
32.2.25	CFDRFCCa : RX FIFO Configuration/Control Registers a (a = 0 to 1).....	1253
32.2.26	CFDRFSTSa : RX FIFO Status Registers a (a = 0 to 1).....	1255
32.2.27	CFDRFPCTRa : RX FIFO Pointer Control Registers a (a = 0 to 1) .....	1256
32.2.28	CFDCFCC : Common FIFO Configuration/Control Register.....	1257
32.2.29	CFDCFSTS : Common FIFO Status Register.....	1260
32.2.30	CFDCFPCTR : Common FIFO Pointer Control Register .....	1262
32.2.31	CFDFESTS : FIFO Empty Status Register .....	1263
32.2.32	CFDFFSTS : FIFO Full Status Register.....	1263
32.2.33	CFDFMSTS : FIFO Message Lost Status Register .....	1264
32.2.34	CFDRFISTS : RX FIFO Interrupt Flag Status Register.....	1265
32.2.35	CFDCDTCT : DMA Transfer Control Register .....	1265
32.2.36	CFDCDTSTS : DMA Transfer Status Register.....	1266
32.2.37	CFDTMCI : TX Message Buffer Control Registers i (i = 0 to 3).....	1267
32.2.38	CFDTMSTSj : TX Message Buffer Status Registers j (j = 0 to 3).....	1268
32.2.39	CFDTMTRSTS : TX Message Buffer Transmission Request Status Register .....	1270
32.2.40	CFDTMTARSTS : TX Message Buffer Transmission Abort Request Status Register .....	1270
32.2.41	CFDTMTCSTS : TX Message Buffer Transmission Completion Status Register .....	1271
32.2.42	CFDTMTASTS : TX Message Buffer Transmission Abort Status Register .....	1271
32.2.43	CFDTMIEC : TX Message Buffer Interrupt Enable Configuration Register.....	1272
32.2.44	CFDTXQCC : TX Queue Configuration/Control Register .....	1273
32.2.45	CFDTXQSTS : TX Queue Status Register .....	1274
32.2.46	CFDTXQPCTR : TX Queue Pointer Control Register.....	1275
32.2.47	CFDTHLCC : TX History List Configuration/Control Register .....	1276
32.2.48	CFDTHLSTS : TX History List Status Register .....	1277
32.2.49	CFDTHLACC0 : TX History List Access Register 0 .....	1278
32.2.50	CFDTHLACC1 : TX History List Access Register 1 .....	1279
32.2.51	CFDTHLPCTR : TX History List Pointer Control Register .....	1280
32.2.52	CFDGRSTC : Global SW reset Register.....	1280
32.2.53	CFDGTSTCFG : Global Test Configuration Register.....	1281

32.2.54	CFDGTSTCTR : Global Test Control Register .....	1282
32.2.55	CFDGFDCFG : Global FD Configuration Register .....	1282
32.2.56	CFDGLCKK : Global Lock Key Register .....	1283
32.2.57	CFDRPGACCK : RAM Test Page Access Registers k (k = 0 to 63) .....	1284
32.2.58	CFDGAFLIGNENT : Global AFL Ignore Entry Register .....	1284
32.2.59	CFDGAFLIGNCTR : Global AFL Ignore Control Register .....	1285
32.2.60	CFDRMIEC : RX Message Buffer Interrupt Enable Configuration Register .....	1285
32.2.61	Message Buffer Component Structure .....	1286
32.3	Modes of Operation .....	1306
32.3.1	Overview .....	1306
32.3.2	Global Modes .....	1306
32.3.3	Channel Modes .....	1315
32.3.4	Global Mode and Channel Mode Transition Interactions .....	1320
32.4	Initialization .....	1322
32.4.1	Initialization of CAN Clock, Bit Timing and Baud Rate .....	1322
32.4.2	CAN Module Configuration after Hardware Reset .....	1329
32.5	Acceptance Filtering Function using Global Acceptance Filter List (AFL) .....	1330
32.5.1	Overview .....	1330
32.5.2	Allocation of AFL Entries .....	1332
32.5.3	AFL Entry Description .....	1332
32.5.4	Entering Entries in the AFL .....	1334
32.5.5	Loopback Modes .....	1336
32.5.6	IDE Masking .....	1336
32.5.7	Updating AFL Entry during Communication .....	1337
32.6	FIFO Buffers and Normal Message Buffer Configuration .....	1339
32.6.1	Normal RX Message Buffers .....	1340
32.6.2	FIFO Buffers .....	1340
32.7	Interrupts and DMA .....	1345
32.7.1	Interrupts .....	1345
32.7.2	DMA Transfer .....	1348
32.8	Reception and Transmission .....	1351
32.8.1	Reception .....	1351
32.8.2	Transmission .....	1357
32.9	Test Mode .....	1371
32.9.1	Channel Specific Test Modes .....	1372
32.9.2	Global Test Modes .....	1374
32.10	RAM Area Configuration .....	1378
32.10.1	Examples .....	1380
32.10.2	OTB Area .....	1381
32.10.3	RAM Initialization Cycle .....	1381



32.11 Usage notes .....	1381
32.11.1 Module-stop function.....	1381
<b>33. CANFD ECC (CNECC) .....</b>	<b>1382</b>
33.1 Overview.....	1382
33.2 Register Descriptions .....	1382
33.2.1 EC710CTL : ECC Control Register.....	1382
33.2.2 EC710TMC : ECC Test Mode Control Register .....	1385
33.2.3 EC710TED : ECC Test Substitute Data Register.....	1386
33.2.4 EC710EAD0 : ECC Error Address Register.....	1386
33.3 Operation.....	1387
33.3.1 <b>ECC Function Setting</b> .....	1387
33.3.2 <b>ECC Decoder Testing</b> .....	1388
33.4 Interrupts .....	1389
<b>34. Serial Peripheral Interface (SPI) .....</b>	<b>1390</b>
34.1 Overview.....	1390
34.2 Register Descriptions .....	1394
34.2.1 SPDR : SPI Data Register .....	1394
34.2.2 SPDECR : SPI Delay Control Register .....	1396
34.2.3 SPCR : SPI Control Register .....	1397
34.2.4 SPCR2 : SPI Control Register 2 .....	1402
34.2.5 SPCR3 : SPI Control Register 3 .....	1404
34.2.6 SPCMDm : SPI Command Register (m = 0 to 7).....	1406
34.2.7 SPDCR : SPI Data Control Register .....	1409
34.2.8 SPDCR2 : SPI Data Control Register 2 .....	1410
34.2.9 SPSR : SPI Status Register .....	1411
34.2.10 SPTFSR : SPI Transfer FIFO Status Register .....	1417
34.2.11 SPRFSR : SPI Receive FIFO Status Register .....	1417
34.2.12 SPPSR : SPI Polling Register .....	1418
34.2.13 SPSRC : SPI Status Clear Register.....	1418
34.2.14 SPFCR : SPI FIFO Clear Register .....	1419
34.3 Operation.....	1419
34.3.1 Overview of SPI Operation.....	1420
34.3.2 Controlling the SPI Pins .....	1421
34.3.3 SPI System Configuration Examples .....	1422
34.3.4 Data Formats .....	1428
34.3.5 Transfer Formats.....	1441
34.3.6 Communication Operating Mode .....	1443
34.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts .....	1446
34.3.8 Idle Interrupt.....	1447

34.3.9	Communication End Interrupt .....	1449
34.3.10	Error Detection .....	1458
34.3.11	Initializing the SPI.....	1465
34.3.12	SPI Operation.....	1465
34.3.13	Clock Synchronous Operation .....	1485
34.3.14	Loopback Mode.....	1491
34.3.15	Self-Diagnosis of Parity Bit Function .....	1492
34.3.16	Interrupt Sources.....	1493
34.4	Event Link Controller Event Output .....	1494
34.4.1	Receive Buffer Full Event Output.....	1494
34.4.2	Transmit Buffer Empty Event Output.....	1494
34.4.3	Mode-Fault, Underrun, Overrun, Parity Error, or received data ready Event Output .....	1495
34.4.4	SPI Idle Event Output.....	1495
34.4.5	Communication End Event Output.....	1496
34.4.6	Synchronization bypass function.....	1498
34.5	Usage Notes.....	1498
34.5.1	Settings for the Module-Stop State .....	1498
34.5.2	Constraint on Low-Power Functions .....	1498
34.5.3	Constraints on Starting Transfer .....	1498
34.5.4	Constraints on Mode-Fault, Underrun, Overrun, Parity Error, or Receive Data Ready Event Output .....	1498
34.5.5	Constraints on the SPSR.SPRF and SPSR.SPTEF Flags .....	1499
<b>35.</b>	<b>Octal Serial Peripheral Interface (OSPI) .....</b>	<b>1500</b>
35.1	Overview.....	1500
35.2	Register Descriptions .....	1501
35.2.1	OSPI Configuration Registers .....	1501
35.2.2	OSPI Control Registers.....	1508
35.2.3	OSPI Status Registers .....	1521
35.2.4	OSPI Interrupt Registers .....	1522
35.3	Operation.....	1526
35.3.1	xSPI Bus .....	1526
35.3.2	Manual-command .....	1538
35.3.3	Memory-mapping .....	1539
35.3.4	Pattern Control .....	1546
35.3.5	Integrity Checking .....	1547
35.3.6	Interrupts .....	1547
35.3.7	Flows of Operations .....	1548
35.3.8	Usage Notes .....	1556
<b>36.</b>	<b>Decryption On The Fly (DOTF) .....</b>	<b>1558</b>
36.1	Overview.....	1558

36.2	Block Diagram .....	1558
36.3	Register Descriptions .....	1559
36.3.1	CONVAREAST : DOTF Conversion Area Start Address Register .....	1559
36.3.2	CONVAREAD : DOTF Conversion Area End Address Register .....	1559
36.4	Operation .....	1560
36.5	System flow .....	1560
36.6	Usage notes .....	1560
36.6.1	Module-stop function .....	1560
<b>37.</b>	<b>Serial Sound Interface Enhanced (SSIE).....</b>	<b>1561</b>
37.1	Overview.....	1561
37.2	Register Descriptions .....	1566
37.2.1	SSICR : Control Register .....	1566
37.2.2	SSISR : Status Register.....	1576
37.2.3	SSIFCR : FIFO Control Register .....	1587
37.2.4	SSIFSR : FIFO Status Register .....	1593
37.2.5	SSIFTDR : Transmit FIFO Data Register .....	1596
37.2.6	SSIFRDR : Receive FIFO Data Register .....	1599
37.2.7	SSIOFR : Audio Format Register .....	1601
37.2.8	SSISCR : Status Control Register.....	1604
37.3	Communication Formats .....	1605
37.3.1	I <sup>2</sup> S Format.....	1606
37.3.2	Monaural Format.....	1606
37.3.3	TDM Format .....	1608
37.4	Communication Modes.....	1609
37.4.1	Slave-mode Communication .....	1609
37.4.2	Master-mode Communication .....	1610
37.4.3	Transmission .....	1610
37.4.4	Reception .....	1610
37.4.5	Transmission and Reception.....	1610
37.5	Operation.....	1610
37.5.1	Idle State .....	1610
37.5.2	Communication States .....	1612
37.6	Communication Operation.....	1616
37.6.1	Start Communication.....	1616
37.6.2	Transmission .....	1617
37.6.3	Reception .....	1618
37.6.4	Transmission and Reception.....	1619
37.6.5	Halt Communication.....	1619
37.6.6	Error Handling .....	1620
37.6.7	Resume Communication.....	1621

37.7	Interrupts .....	1622
37.7.1	SSIE <sub>n</sub> _SSIF Interrupt (n = 0, 1) .....	1623
37.7.2	SSIE <sub>0</sub> _SSITXI Interrupt [Full-duplex communication] .....	1624
37.7.3	SSIE <sub>0</sub> _SSIRXI Interrupt [Full-duplex communication] .....	1624
37.7.4	SSIE <sub>1</sub> _SSIRT Interrupt [Half-duplex communication] .....	1624
37.8	Software Resets .....	1625
37.8.1	Software Reset Procedure .....	1625
37.9	Usage Notes .....	1626
37.9.1	Notes for Slave-mode Communication .....	1626
37.9.2	Notes for Master-mode Communication .....	1627
37.9.3	Notes for Communication Flow .....	1627
37.9.4	Write Access Restriction .....	1628
37.9.5	Module-stop function .....	1629
<b>38.</b>	<b>Cyclic Redundancy Check (CRC) .....</b>	<b>1630</b>
38.1	Overview .....	1630
38.2	Register Descriptions .....	1631
38.2.1	CRCCR0 : CRC Control Register 0 .....	1631
38.2.2	CRCCR1 : CRC Control Register 1 .....	1631
38.2.3	CRCDIR/CRCDIR_BY : CRC Data Input Register .....	1632
38.2.4	CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register .....	1632
38.2.5	CRCSAR : Snoop Address Register .....	1633
38.3	Operation .....	1634
38.3.1	Basic Operation .....	1634
38.3.2	CRC Snoop Function .....	1637
38.4	Usage Notes .....	1638
38.4.1	Settings for the Module-Stop State .....	1638
38.4.2	Note on Transmission .....	1638
<b>39.</b>	<b>Boundary Scan .....</b>	<b>1640</b>
39.1	Overview .....	1640
39.2	Register Descriptions .....	1640
39.2.1	JTIR : Instruction Register .....	1641
39.2.2	JTIDR : ID Code Register .....	1641
39.2.3	JTBPR : Bypass Register .....	1642
39.2.4	JTBSR : Boundary Scan Register .....	1642
39.3	Operation .....	1642
39.3.1	TAP Controller .....	1642
39.3.2	Commands .....	1643
39.4	Usage Notes .....	1644
<b>40.</b>	<b>Security Features .....</b>	<b>1646</b>

40.1	Features .....	1646
40.2	Tamper Detection .....	1647
40.3	Arm Security Features .....	1647
40.3.1	Arm TrustZone Technology .....	1647
40.3.2	Privileged Control .....	1648
40.3.3	Security Attribution .....	1648
40.3.4	TrustZone Access Error .....	1653
40.4	Device Lifecycle Management .....	1654
40.4.1	Changing the Lifecycle State .....	1655
40.4.2	Protection and Authentication Level .....	1655
40.4.3	Serial Programming .....	1656
40.4.4	Device Lifecycle State and PL Change Example .....	1657
40.4.5	Failure Analysis .....	1657
40.5	Secure Key Injection .....	1657
40.6	Secure Factory Programming .....	1659
40.7	Field Updating in Dual Mode .....	1660
40.8	Register Description .....	1662
40.8.1	PSARB : Peripheral Security Attribution Register B .....	1662
40.8.2	PSARC : Peripheral Security Attribution Register C .....	1663
40.8.3	PSARD : Peripheral Security Attribution Register D .....	1664
40.8.4	PSARE : Peripheral Security Attribution Register E .....	1666
40.8.5	MSSAR : Module Stop Security Attribution Register .....	1667
40.8.6	PPARB : Peripheral Privilege Attribution Register B .....	1668
40.8.7	PPARC : Peripheral Privilege Attribution Register C .....	1670
40.8.8	PPARD : Peripheral Privilege Attribution Register D .....	1671
40.8.9	PPARE : Peripheral Privilege Attribution Register E .....	1672
40.8.10	MSPAR : Module Stop Privilege Attribution Register .....	1674
40.8.11	CFSAMONA : Code Flash Security Attribution Monitor Register A .....	1674
40.8.12	DFSAMON : Data Flash Security Attribution Monitor Register .....	1674
40.8.13	DLMMON : Device Lifecycle Management State Monitor Register .....	1675
40.8.14	MSAOAD : Master Security Attribution Operation After Detection Register .....	1675
40.8.15	MSAPT : Master Security Attribution Protect Register .....	1676
40.9	Usage Note .....	1676
40.9.1	Security or Privilege Bit Write Timing .....	1676
<b>41.</b>	<b>Renesas Secure IP (RSIP-E51A) .....</b>	<b>1677</b>
41.1	Overview .....	1677
41.2	Operation .....	1679
41.2.1	Symmetric cryptography .....	1679
41.2.2	Asymmetric Cryptography .....	1680
41.2.3	Hash generator .....	1681

41.2.4	Encryption and Decryption .....	1681
41.3	Usage Notes .....	1682
41.3.1	Software Standby Mode .....	1682
41.3.2	Module-Stop Function Setting .....	1682
41.3.3	SPA/DPA Protections .....	1682
41.3.4	Restrictions on Open process .....	1682
<b>42.</b>	<b>12-Bit A/D Converter (ADC12) .....</b>	<b>1683</b>
42.1	Overview .....	1683
42.2	Register Descriptions .....	1688
42.2.1	ADDRn : A/D Data Registers n .....	1688
42.2.2	ADDBLDR : A/D Data Duplexing Register .....	1689
42.2.3	ADDBLDRn : A/D Data Duplexing Register n (n = A, B) .....	1690
42.2.4	ADTSDR : A/D Temperature Sensor Data Register .....	1692
42.2.5	ADOCDR : A/D Internal Reference Voltage Data Register .....	1693
42.2.6	ADVMDR : A/D VBATT Monitor Data Register .....	1694
42.2.7	ADRD : A/D Self-Diagnosis Data Register .....	1695
42.2.8	ADCSR : A/D Control Register .....	1696
42.2.9	ADANSA0 : A/D Channel Select Register A0 .....	1700
42.2.10	ADANSB0 : A/D Channel Select Register B0 .....	1700
42.2.11	ADADS0 : A/D-Converted Value Addition/Average Channel Select Register 0 .....	1701
42.2.12	ADADC : A/D-Converted Value Addition/Average Count Select Register .....	1702
42.2.13	ADCER : A/D Control Extended Register .....	1703
42.2.14	ADSTRGR : A/D Conversion Start Trigger Select Register .....	1705
42.2.15	ADEXICR : A/D Conversion Extended Input Control Registers .....	1707
42.2.16	ADSSTRn/ADSSTRT/ADSSTRO/ADSSTRV: A/D Sampling State Register (n = 0 to 8) .....	1709
42.2.17	ADSHCR : A/D Sample and Hold Circuit Control Register .....	1709
42.2.18	ADSHMSR : A/D Sample and Hold Operation Mode Selection Register .....	1710
42.2.19	ADDISCR : A/D Disconnection Detection Control Register .....	1711
42.2.20	ADGSPCR : A/D Group Scan Priority Control Register .....	1711
42.2.21	ADCMPCR : A/D Compare Function Control Register .....	1713
42.2.22	ADCMPANSR0 : A/D Compare Function Window A Channel Select Register 0 .....	1714
42.2.23	ADCMPANSER : A/D Compare Function Window A Extended Input Select Register .....	1715
42.2.24	ADCMPLR0 : A/D Compare Function Window A Comparison Condition Setting Register 0 .....	1715
42.2.25	ADCMPLER : A/D Compare Function Window A Extended Input Comparison Condition Setting Register .....	1717
42.2.26	ADCMPLRn : A/D Compare Function Window A Lower-Side/Upper-Side Level Setting Register (n = 0, 1) .....	1718
42.2.27	ADWINnLB : A/D Compare Function Window B Lower-Side/Upper-Side Level Setting Register (n = L, U) .....	1720
42.2.28	ADCMPSR0 : A/D Compare Function Window A Channel Status Register 0 .....	1721

42.2.29	ADCMPSER : A/D Compare Function Window A Extended Input Channel Status Register .....	1722
42.2.30	ADCMPBNSR : A/D Compare Function Window B Channel Select Register .....	1723
42.2.31	ADCMPBSR : A/D Compare Function Window B Status Register .....	1724
42.2.32	ADWINMON : A/D Compare Function Window A/B Status Monitor Register .....	1725
42.2.33	ADBUFEN : A/D Data Buffer Enable Register .....	1726
42.2.34	ADBUFPTR : A/D Data Buffer Pointer Register .....	1727
42.2.35	ADBUFn : A/D Data Buffer Registers n (n = 0 to 15) .....	1727
42.3	Operation .....	1728
42.3.1	Scanning Operation .....	1728
42.3.2	Single Scan Mode .....	1729
42.3.3	Continuous Scan Mode .....	1738
42.3.4	Group Scan Mode .....	1745
42.3.5	Compare Function for Windows A and B .....	1754
42.3.6	Analog Input Sampling and Scan Conversion Time .....	1758
42.3.7	Usage Example of A/D Data Register Automatic Clearing Function .....	1762
42.3.8	A/D-Converted Value Addition/Average Mode .....	1762
42.3.9	Disconnection Detection Assist Function .....	1762
42.3.10	Starting A/D Conversion with an Asynchronous Trigger .....	1764
42.3.11	Starting A/D Conversion with a Synchronous Trigger from a Peripheral Module .....	1765
42.3.12	Security Attribution between ADCn (n = 0, 1) and PORT (PmSAR (m = 0 to 9)) .....	1765
42.3.13	Using Data Buffers .....	1765
42.4	Interrupt Sources and DTC, DMAC Transfer Requests .....	1766
42.4.1	Interrupt Requests .....	1766
42.5	Event Link Function .....	1768
42.5.1	Event Output to the ELC .....	1768
42.5.2	ADC12 Operation through an Event from the ELC .....	1768
42.6	Usage Notes .....	1769
42.6.1	Constraints on Setting the Registers .....	1769
42.6.2	Constraints on Reading the Data Registers .....	1769
42.6.3	Constraints on Stopping A/D Conversion .....	1769
42.6.4	A/D Conversion Restart and Termination Timing .....	1771
42.6.5	Constraints on Scan End Interrupt Handling .....	1771
42.6.6	Settings for the Module-Stop Function .....	1771
42.6.7	Notes on Entering the Low-Power States .....	1771
42.6.8	Notes on Canceling Software Standby Mode .....	1771
42.6.9	Error in Absolute Accuracy When Disconnection Detection Assistance Is in Use .....	1771
42.6.10	Constraints on Operating Modes and Status Bits .....	1772
42.6.11	Notes on Board Design .....	1772
42.6.12	Constraints on Noise Prevention .....	1772

42.6.13	Port Settings When Using the ADC12 Input .....	1773
42.6.14	Relationship between ADC12 Units 0 and 1 and the ACMPHS.....	1773
42.6.15	Calculation for Sampling Time .....	1773
42.6.16	Notes on simultaneous operation with other modules .....	1774
<b>43.</b>	<b>12-Bit D/A Converter (DAC12) .....</b>	<b>1775</b>
43.1	Overview.....	1775
43.2	Register Descriptions .....	1776
43.2.1	DADRn : D/A Data Register n (n = 0).....	1776
43.2.2	DACR : D/A Control Register .....	1776
43.2.3	DADPR : DADRn Format Select Register.....	1777
43.2.4	DAADSCR : D/A A/D Synchronous Start Control Register .....	1777
43.2.5	DAAMPCR : D/A Output Amplifier Control Register .....	1778
43.2.6	DAASWCR : D/A Amplifier Stabilization Wait Control Register .....	1778
43.2.7	DAADUSR : D/A A/D Synchronous Unit Select Register.....	1779
43.3	Operation.....	1779
43.3.1	Reducing Interference between D/A and A/D Conversion .....	1780
43.4	Event Link Operation Setting Procedure .....	1782
43.4.1	DA0 Event Link Operation Setting Procedure.....	1782
43.5	Analog Output Control by Security Attribution .....	1782
43.6	Usage Notes on Event Link Operation .....	1783
43.7	Usage Notes.....	1783
43.7.1	Settings for the Module-Stop Function.....	1783
43.7.2	DAC12 Operation in the Module-Stop State .....	1783
43.7.3	DAC12 Operation in Software Standby Mode.....	1783
43.7.4	Constraint on Entering Deep Software Standby Mode .....	1783
43.7.5	Initialization Procedure with the Output Amplifier.....	1783
43.7.6	Initialization Procedure of the Output to internal modules.....	1784
43.7.7	Constraint on Usage When Interference Reduction between D/A and A/D Conversion Is Enabled .....	1784
<b>44.</b>	<b>Temperature Sensor (TSN) .....</b>	<b>1785</b>
44.1	Overview.....	1785
44.2	Register Descriptions .....	1786
44.2.1	TSCR : Temperature Sensor Control Register.....	1786
44.2.2	TSCDR : Temperature Sensor Calibration Data Register .....	1786
44.3	Using the Temperature Sensor.....	1787
44.3.1	Preparation for Using the Temperature Sensor.....	1787
44.3.2	Procedures for Using the Temperature Sensor.....	1788
44.4	Usage Notes.....	1790
44.4.1	Settings for the Module-Stop Function.....	1790
44.4.2	Settings for the Software Standby Mode .....	1790



<b>45. High-Speed Analog Comparator (ACMPHS)</b> .....	<b>1791</b>
45.1 Overview.....	1791
45.2 Register Descriptions .....	1792
45.2.1 CMPCTL : Comparator Control Register .....	1792
45.2.2 CMPSEL0 : Comparator Input Select Register .....	1793
45.2.3 CMPSEL1 : Comparator Reference Voltage Select Register .....	1794
45.2.4 CMPMON : Comparator Output Monitor Register.....	1794
45.2.5 CPIOC : Comparator Output Control Register .....	1795
45.2.6 CPINTCTL : Comparator Interrupt Control Register .....	1795
45.2.7 CPMSKCTL : Comparator Interrupt Mask Control Register.....	1796
45.3 Operation.....	1796
45.4 Noise Filter .....	1798
45.5 ACMPHS Interrupts .....	1799
45.6 ACMPHS Output to the Event Link Controller (ELC).....	1799
45.7 ACMPHS Pin Output .....	1799
45.8 Usage Notes.....	1800
45.8.1 Settings for the Module-Stop Function.....	1800
45.8.2 Settings for the DAC12 .....	1800
45.8.3 Relationship with the ADC12 .....	1800
45.8.4 ACMPHS Operation in Module-Stop State .....	1800
45.8.5 ACMPHS Operation in Software Standby Mode.....	1800
45.8.6 Setting the D/A Converter for Generating Reference Voltage.....	1800
<b>46. Data Operation Circuit (DOC)</b> .....	<b>1801</b>
46.1 Overview.....	1801
46.2 DOC Register Descriptions .....	1802
46.2.1 DOCR : DOC Control Register.....	1802
46.2.2 DOSR : DOC Flag Status Register .....	1803
46.2.3 DOSCR : DOC Flag Status Clear Register .....	1804
46.2.4 DODIR : DOC Data Input Register.....	1804
46.2.5 DODSR0 : DOC Data Setting Register 0 .....	1804
46.2.6 DODSR1 : DOC Data Setting Register 1 .....	1805
46.3 Operation.....	1805
46.3.1 Data Comparison Mode .....	1805
46.3.2 Data Addition Mode.....	1807
46.3.3 Data Subtraction Mode .....	1808
46.4 Interrupt Source.....	1808
46.5 Event Link Output.....	1809
46.6 Interrupt Handling and Event Linking .....	1809
46.7 Usage Notes.....	1809
46.7.1 Settings for the Module-Stop State .....	1809

<b>47. SRAM</b> .....	<b>1810</b>
47.1 Overview.....	1810
47.2 Register Descriptions .....	1810
47.2.1 SRAMSABAR1 : SRAM Error Address Register .....	1810
47.2.2 SRAMSAR : SRAM Security Attribution Register .....	1811
47.2.3 SRAMPRCR_S : SRAM Protection Control Register for Secure .....	1811
47.2.4 SRAMPRCR_NS : SRAM Protection Control Register for Non-secure .....	1812
47.2.5 SRAMWTSC : SRAM Wait State Control Register .....	1813
47.2.6 SRAMCR1 : SRAM Control Register 1 .....	1814
47.2.7 SRAMESR : SRAM Error Status Register .....	1815
47.2.8 SRAMESCLR : SRAM Error Status Clear Register .....	1815
47.2.9 SRAMEAR2 : SRAM Error Address Register .....	1816
47.3 Operation.....	1816
47.3.1 Module Stop Function .....	1816
47.3.2 Parity Calculation Function .....	1817
47.3.3 TrustZone Filter function .....	1818
47.3.4 Interrupt Source .....	1819
47.3.5 Wait state .....	1819
47.4 Usage Note.....	1819
47.4.1 Instruction fetch from SRAM area .....	1819
47.4.2 Note of write to SRAMCR1 register .....	1819
47.4.3 Notes on Using Error Checking of SRAM .....	1820
<b>48. Standby SRAM</b> .....	<b>1821</b>
48.1 Overview.....	1821
48.2 Register Descriptions .....	1821
48.2.1 STBRAMSABAR : Standby SRAM Security Attribute Boundary Address Register .....	1821
48.2.2 STBRAMPABAR_NS : Standby SRAM Privilege Attribute Boundary Address Register for Non-secure.....	1822
48.2.3 STBRAMPABAR_S : Standby SRAM Privilege Attribute Boundary Address Register for Secure .....	1822
48.2.4 STBRAMCR : Standby SRAM Control Register .....	1823
48.2.5 STBRAMEAR : Standby SRAM Error Address Register.....	1823
48.3 Operation.....	1823
48.3.1 Data Retention .....	1823
48.3.2 Module-stop Function.....	1824
48.3.3 Parity Calculation Function .....	1824
48.3.4 TrustZone Filter function .....	1826
48.3.5 Interrupt Source .....	1827
48.3.6 Wait state .....	1827
48.4 Usage Notes.....	1828

48.4.1	Instruction Fetch from the Standby SRAM Area .....	1828
48.4.2	Notes on Self-Diagnosis of the Standby SRAM .....	1828
<b>49.</b>	<b>Flash Memory .....</b>	<b>1829</b>
49.1	Overview.....	1829
49.2	Structure of Memory .....	1831
49.3	Address Space .....	1834
49.4	Register Descriptions .....	1834
49.4.1	FCACHEE : Flash Cache Enable Register .....	1834
49.4.2	FCACHEIV : Flash Cache Invalidate Register .....	1835
49.4.3	FLWT : Flash Wait Cycle Register .....	1835
49.4.4	FSAR : Flash Security Attribution Register .....	1836
49.4.5	UIDRn : Unique ID Registers n (n = 0 to 3).....	1837
49.4.6	PNRn : Part Numbering Register n (n = 0 to 3).....	1837
49.4.7	MCUVER : MCU Version Register .....	1837
49.4.8	FWEPROR : Flash P/E Protect Register .....	1838
49.4.9	FASTAT : Flash Access Status Register .....	1838
49.4.10	FAEINT : Flash Access Error Interrupt Enable Register .....	1840
49.4.11	FRDYIE : Flash Ready Interrupt Enable Register .....	1840
49.4.12	FSADDR : FACI Command Start Address Register.....	1841
49.4.13	FEADDR : FACI Command End Address Register.....	1842
49.4.14	FMEPROT : Flash P/E Mode Entry Protection Register .....	1842
49.4.15	FCNTSELR : Flash Counter Select Register .....	1843
49.4.16	FCNTDATARn : Flash Counter Data Register n (n = 0, 1).....	1844
49.4.17	FBPROT0 : Flash Block Protection Register .....	1845
49.4.18	FBPROT1 : Flash Block Protection for Secure Register.....	1845
49.4.19	FSTATR : Flash Status Register .....	1846
49.4.20	FENTRYR : Flash P/E Mode Entry Register .....	1850
49.4.21	FSUINTR : Flash Sequencer Setup Initialization Register .....	1851
49.4.22	FCMDR : FACI Command Register .....	1852
49.4.23	FBCCNT : Blank Check Control Register .....	1853
49.4.24	FBCSTAT : Blank Check Status Register.....	1853
49.4.25	FPSADDR : Data Flash Programming Start Address Register .....	1854
49.4.26	FSUASMON : Flash Startup Area Select Monitor Register .....	1854
49.4.27	FCPSR : Flash Sequencer Processing Switching Register .....	1855
49.4.28	FPCKAR : Flash Sequencer Processing Clock Notification Register .....	1856
49.4.29	FSUACR : Flash Startup Area Control Register .....	1856
49.4.30	FCKMHZ : Data Flash Access Frequency Register .....	1857
49.5	Flash Cache .....	1858
49.5.1	Feature of flash cache.....	1858
49.6	Operating Modes Associated with Flash Memory .....	1859

49.7	Overview of Functions .....	1860
49.8	Operating Modes of the Flash Sequencer .....	1861
49.9	FACI Commands .....	1862
49.9.1	List of FACI Commands .....	1862
49.9.2	Relationship between the Flash Sequencer State and FACI Commands .....	1864
49.9.3	Usage of FACI Commands .....	1865
49.10	Suspend Operation .....	1888
49.11	Protection Function .....	1888
49.11.1	Software Protection .....	1888
49.11.2	Error Protection .....	1892
49.11.3	Start-Up Program Protection .....	1895
49.11.4	Dual Bank Function .....	1899
49.11.5	Block Swap Function .....	1901
49.12	Security Function .....	1902
49.12.1	Security Flag for Startup Area Select .....	1902
49.12.2	Permanent Block Protect Setting .....	1902
49.12.3	Flash Memory Protection for TrustZone .....	1904
49.12.4	Data Flash Configuration Area Protection .....	1908
49.12.5	Anti-Rollback Counter .....	1909
49.13	Boot Mode .....	1913
49.13.1	Boot Mode (for the SCI Interface) .....	1914
49.13.2	Boot Mode (for the USB Interface) .....	1914
49.14	Using the Serial Programmer for Rewriting .....	1915
49.14.1	Environments for Serial Programming .....	1915
49.15	Programming through Self-Programming .....	1916
49.15.1	Overview .....	1916
49.15.2	Background Operation .....	1917
49.16	Reading Flash Memory .....	1917
49.16.1	Reading Code Flash Memory .....	1917
49.16.2	Reading Data Flash Memory .....	1918
49.17	Usage Notes .....	1918
<b>50.</b>	<b>Capture Engine Unit (CEU) .....</b>	<b>1920</b>
50.1	Overview .....	1920
50.2	Register Descriptions .....	1923
50.2.1	CAPSR : Capture Start Register .....	1924
50.2.2	CAPCR : Capture Control Register .....	1928
50.2.3	CAMCR : Capture Interface Control Register .....	1930
50.2.4	CMCYR : Capture Interface Cycle Register .....	1934
50.2.5	CAMOR, CAMOR_x : Capture Interface Offset Register (x = B, M) .....	1935
50.2.6	CAPWR, CAPWR_x : Capture Interface Width Register (x = B, M) .....	1936

50.2.7	CAIFR : Capture Interface Input Format Register .....	1938
50.2.8	CRCNTR : CEU Register Control Register .....	1943
50.2.9	CRCMPR : CEU Register Forcible Control Register .....	1944
50.2.10	CFLCR, CFLCR_x : Capture Filter Control Register (x = B, M) .....	1945
50.2.11	CFSZR, CFSZR_x : Capture Filter Size Clip Register (x = B, M) .....	1947
50.2.12	CDWDR, CDWDR_x : Capture Destination Width Register (x = B, M) .....	1948
50.2.13	CDAYR, CDAYR_x : Capture Data Address Y Register (x = B, M) .....	1949
50.2.14	CDACR, CDACR_x : Capture Data Address C Register (x = B, M) .....	1950
50.2.15	CDBYR, CDBYR_x : Capture Data Bottom-Field Address Y Register (x = B, M) .....	1952
50.2.16	CDBCR, CDBCR_x : Capture Data Bottom-Field Address C Register (x = B, M) .....	1953
50.2.17	CBDSR, CBDSR_x : Capture Bundle Destination Size Register (x = B, M) .....	1954
50.2.18	CFWCR : Firewall Operation Control Register .....	1955
50.2.19	CLFCR, CLFCR_x : Capture Low-Pass Filter Control Register (x = B, M) .....	1956
50.2.20	CDOCR, CDOCR_x : Capture Data Output Control Register (x = B, M) .....	1956
50.2.21	CEIER : Capture Event Interrupt Enable Register .....	1962
50.2.22	CETCR : Capture Event Flag Clear Register .....	1963
50.2.23	CSTSR : Capture Status Register .....	1967
50.2.24	CDSSR : Capture Data Size Register .....	1968
50.2.25	CDAYR2, CDAYR2_x : Capture Data Address Y Register 2 (x = B, M) .....	1970
50.2.26	CDACR2, CDACR2_x : Capture Data Address C Register 2 (x = B, M) .....	1971
50.2.27	CDBYR2, CDBYR2_x : Capture Data Bottom-Field Address Y Register 2 (x = B, M) .....	1973
50.2.28	CDBCR2, CDBCR2_x : Capture Data Bottom-Field Address C Register 2 (x = B, M) .....	1974
50.2.29	CBWER : CEU Bufferable Write Enable Register .....	1975
50.3	Usage Notes .....	1975
50.3.1	Settings for the Module-Stop Function .....	1975
50.3.2	Conditions for Connection to an External Module .....	1976
50.3.3	Restrictions on Input/Output Functions .....	1976
<b>51.</b>	<b>Internal Voltage Regulator .....</b>	<b>1977</b>
51.1	Overview .....	1977
51.2	Operation .....	1977
51.2.1	DCDC mode .....	1977
51.2.2	External VDD mode .....	1978
51.3	Usage Notes .....	1980
<b>52.</b>	<b>Electrical Characteristics .....</b>	<b>1981</b>
52.1	Absolute Maximum Ratings .....	1981
52.2	DC Characteristics .....	1982
52.2.1	T <sub>j</sub> /T <sub>a</sub> Definition .....	1982
52.2.2	I/O V <sub>IH</sub> , V <sub>IL</sub> .....	1983
52.2.3	I/O I <sub>OH</sub> , I <sub>OL</sub> .....	1985

52.2.4	I/O $V_{OH}$ , $V_{OL}$ , and Other Characteristics.....	1987
52.2.5	Operating and Standby Current .....	1988
52.2.6	VCC Rise and Fall Gradient and Ripple Frequency.....	2002
52.2.7	Thermal Characteristics .....	2003
52.3	AC Characteristics.....	2007
52.3.1	Frequency .....	2007
52.3.2	Clock Timing.....	2008
52.3.3	Reset Timing .....	2011
52.3.4	Wakeup Timing.....	2012
52.3.5	NMI and IRQ Noise Filter .....	2017
52.3.6	I/O Ports, POEG, GPT, AGT, ULPT and ADC12 Trigger Timing .....	2018
52.3.7	CAC Timing .....	2022
52.3.8	SCI Timing.....	2023
52.3.9	SPI Timing.....	2033
52.3.10	OSPI Timing .....	2041
52.3.11	IIC Timing .....	2045
52.3.12	SSIE Timing .....	2048
52.3.13	ETHERC Timing.....	2052
52.3.14	CEU Timing .....	2056
52.3.15	CANFD Timing .....	2058
52.4	USB Characteristics .....	2058
52.4.1	USBFS Timing.....	2058
52.5	ADC12 Characteristics .....	2060
52.6	DAC12 Characteristics .....	2065
52.7	TSN Characteristics.....	2066
52.8	OSC Stop Detect Characteristics .....	2066
52.9	POR and PVD Characteristics.....	2067
52.10	VBATT Characteristics .....	2070
52.11	ACMPHS Characteristics .....	2071
52.12	Flash Memory Characteristics .....	2072
52.12.1	Code Flash Memory Characteristics .....	2072
52.12.2	Data Flash Memory Characteristics .....	2073
52.12.3	Option Setting Memory (Code flash memory) Characteristics .....	2074
52.12.4	Option Setting Memory (Data flash memory) Characteristics .....	2075
52.12.5	Anti-rollback counter Characteristics.....	2075
52.13	Boundary Scan .....	2076
52.14	Joint European Test Action Group (JTAG) .....	2078
52.15	Serial Wire Debug (SWD).....	2080
52.16	Embedded Trace Macro Interface (ETM) .....	2081

<b>Appendix 1. Port States in Each Processing Mode.....</b>	<b>2083</b>
---	-------------

<b>Appendix 2. Package Dimensions .....</b>	<b>2084</b>
<b>Appendix 3. I/O Registers .....</b>	<b>2086</b>
3.1 Peripheral Base Addresses .....	2086
3.2 Access Cycles .....	2088
<b>Appendix 4. Note for Register R/W.....</b>	<b>2091</b>
<b>Appendix 5. Peripheral Variant .....</b>	<b>2092</b>
<b>Revision History .....</b>	<b>2093</b>

## Renesas RA8E1 Group User's Manual

High-performance 360 MHz Arm® Cortex®-M85 core with Helium™, 1 MB code flash memory with Dual-bank, background and SWAP operation, 12 KB Data flash memory, and 544 KB SRAM with Parity. High-integration with Ethernet MAC controller, USB 2.0 Full-Speed, CANFD, Octal SPI and advanced analog.

## Features

- **Arm® Cortex®-M85 core with Helium™**
  - Armv8.1-M architecture profile
  - Armv8-M Security Extension
  - Maximum operating frequency: 360 MHz
  - Memory Protection Unit (Arm MPU)
    - Protected Memory System Architecture (PMSAv8)
    - Secure MPU (MPU\_S): 8 regions
    - Non-secure MPU (MPU\_NS): 8 regions
  - SysTick timer
    - Embeds two SysTick timers: Secure and Non-secure instance
    - Driven by CPUCLK or MOCO divided by 8
  - CoreSight™ ETM-M85
- **Memory**
  - 1 MB code flash memory
  - 12 KB data flash memory (100,000 program/erase (P/E) cycles)
  - 544 KB SRAM including 32 KB of TCM
- **Connectivity**
  - Serial Communications Interface (SCI) × 6, up to 60 Mbps
    - Asynchronous interfaces
    - 8 bit clock synchronous interface
    - Smart card interface
    - Simple IIC
    - Simple SPI
    - Manchester coding (SCI0)
    - Simple LIN (SCI0, SCI1)
  - I<sup>2</sup>C bus interface (IIC) × 2
  - Serial Peripheral Interface (SPI) × 2, up to 60 Mbps
  - Octal Serial Peripheral Interface (OSPI)
  - USB 2.0 Full-Speed Module (USBFS)
  - CAN with Flexible Data-rate (CANFD) × 2
  - Ethernet MAC/DMA Controller (ETHERC/EDMAC)
  - Serial Sound Interface Enhanced (SSIE) × 2
- **Analog**
  - 12-bit A/D Converter (ADC12) × 2
  - 12-bit D/A Converter (DAC12)
  - High-Speed Analog Comparator (ACMPHS) × 2
  - Temperature Sensor (TSN)
- **Timers**
  - General PWM Timer 32-bit (GPT32) × 6
  - General PWM Timer 16-bit (GPT16) × 4
  - Low Power Asynchronous General Purpose Timer (AGT) × 2
  - Ultra-Low-Power Timer (ULPT) × 2
- **Security and Encryption**
  - Renesas Secure IP (RSIP-E51A)
    - 128 bit unique ID
  - Arm® TrustZone®
    - Up to two or four regions for the code flash, depending on the bank mode
    - Up to two regions for the data flash
    - Up to two regions for the SRAM
    - Individual Secure or Non-secure security attribution for each peripheral
  - Privileged control
  - Device lifecycle management
  - Pin function
    - Up to three tamper-resistant pins
    - Secure pin multiplexing
- **System and Power Management**
  - Low power modes
  - Battery backup function (VBATT)
  - Realtime Clock (RTC) with calendar and VBATT support
  - Event Link Controller (ELC)
  - Data Transfer Controller (DTC)
  - DMA Controller (DMAC) × 8
  - Power-on reset
  - Programmable Voltage Detection (PVD) with voltage settings
- Watchdog Timer (WDT)
- Independent Watchdog Timer (IWDT)
- **Human Machine Interface (HMI)**
  - Capture Engine Unit (CEU)
- **Multiple Clock Sources**
  - Main clock oscillator (MOSC) (8 to 48 MHz)
  - Sub-clock oscillator (SOSC) (32.768 kHz)
  - High-speed on-chip oscillator (HOCO) (16/18/20/32/48 MHz)
  - Middle-speed on-chip oscillator (MOCO) (8 MHz)
  - Low-speed on-chip oscillator (LOCO) (32.768 kHz)
  - Clock trim function for HOCO/MOCO/LOCO
  - PLL1/PLL2
  - Clock out support
- **General-Purpose I/O Ports**
  - 5-V tolerance, open drain, input pull-up, switchable driving ability
- **Operating Voltage**
  - VCC: 1.68 to 3.6 V
  - VCC2: 1.65 to 3.6 V
- **Operating Junction Temperature and Packages**
  - T<sub>j</sub> = -40°C to +105°C
    - 100-pin LQFP (14 mm × 14 mm, 0.5 mm pitch)
    - 144-pin LQFP (20 mm × 20 mm, 0.5 mm pitch)



## 1. Overview

The MCU integrates multiple series of software-compatible Arm®-based 32-bit cores that share a common set of Renesas peripherals to facilitate design scalability and efficient platform-based product development.

The MCU in this series incorporates a high-performance Arm® Cortex®-M85 core with Helium™ running up to 360 MHz with the following features:

- 1 MB code flash memory
- 544 KB SRAM (32 KB of TCM RAM, 512 KB of user SRAM)
- Octal Serial Peripheral Interface (OSPI)
- Ethernet MAC Controller (ETHERC), USBFS
- Analog peripherals
- Security and safety features

### 1.1 Function Outline

**Table 1.1 Arm core**

Feature	Functional description
Arm® Cortex®-M85 core	<ul style="list-style-type: none"> <li>• Maximum operating frequency: up to 360 MHz</li> <li>• Arm® Cortex®-M85 core               <ul style="list-style-type: none"> <li>– Revision: (r0p2-00rel0)</li> <li>– ARMv8.1-M architecture profile</li> <li>– Armv8-M Security Extension</li> <li>– Floating Point Unit (FPU) compliant with the ANSI/IEEE Std 754-2008 Scalar half, single, and double-precision floating-point operation</li> <li>– M-profile Vector Extension (MVE) Integer, half-precision, and single-precision floating-point MVE (MVE-F)</li> <li>– Helium™ technology is M-profile Vector Extension (MVE)</li> </ul> </li> <li>• Arm® Memory Protection Unit (Arm MPU)               <ul style="list-style-type: none"> <li>– Protected Memory System Architecture (PMSAv8)</li> <li>– Secure MPU (MPU_S): 8 regions</li> <li>– Non-secure MPU (MPU_NS): 8 regions</li> </ul> </li> <li>• SysTick timer               <ul style="list-style-type: none"> <li>– Embeds two SysTick timers: Secure instance (SysTick_S) and Non-secure instance (SysTick_NS)</li> <li>– Driven by CPUCLK or MOCO divided by 8</li> </ul> </li> <li>• CoreSight™ ETM-M85</li> </ul>

**Table 1.2 Memory**

Feature	Functional description
Code flash memory	1 MB of code flash memory. See <a href="#">section 49, Flash Memory</a> .
Data flash memory	12 KB of data flash memory. See <a href="#">section 49, Flash Memory</a> .
Option-setting memory	The option-setting memory determines the state of the MCU after a reset. See <a href="#">section 6, Option-Setting Memory</a> .
SRAM	On-chip high-speed SRAM with parity bit. See <a href="#">section 47, SRAM</a> .
Standby SRAM	On-chip SRAM that can retain data in Deep Software Standby mode 1. See <a href="#">section 48, Standby SRAM</a> .

**Table 1.3 System**

Feature	Functional description
Operating modes	Three operating modes: <ul style="list-style-type: none"> <li>• Single-chip mode</li> <li>• JTAG boot mode</li> <li>• SCI/USB boot mode</li> </ul> See <a href="#">section 3, Operating Modes</a> .
Resets	This MCU provides 13 types of reset. See <a href="#">section 5, Resets</a> .
Programmable Voltage Detection (PVD)	The Programmable Voltage Detection (PVD) module monitors the voltage level input to the VCC pin. The detection level can be selected by register settings. The PVD module consists of three separate voltage level detectors (PVD0, PVD1, PVD2). PVD0, PVD1, and PVD2 measure the voltage level input to the VCC pin. PVD registers allow your application to configure detection of VCC changes at various voltage thresholds. See <a href="#">section 7, Programmable Voltage Detection (PVD)</a> .
Clocks	<ul style="list-style-type: none"> <li>• Main clock oscillator (MOSC)</li> <li>• Sub-clock oscillator (SOSC)</li> <li>• High-speed on-chip oscillator (HOCO)</li> <li>• Middle-speed on-chip oscillator (MOCO)</li> <li>• Low-speed on-chip oscillator (LOCO)</li> <li>• PLL1/PLL2</li> <li>• Clock out support</li> </ul> See <a href="#">section 8, Clock Generation Circuit</a> .
Clock Frequency Accuracy Measurement Circuit (CAC)	The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated. See <a href="#">section 9, Clock Frequency Accuracy Measurement Circuit (CAC)</a> .
Interrupt Controller Unit (ICU)	The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), the DMA Controller (DMAC), and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts. See <a href="#">section 13, Interrupt Controller Unit (ICU)</a> .
Low power modes	Power consumption can be reduced in multiple ways, including setting clock dividers, stopping modules, power gating control, selecting operating power control modes in normal operation, and transitioning to low power modes and processor low power modes. See <a href="#">section 10, Low Power Modes</a> .
Battery backup function	A battery backup function is provided for partial powering by a battery. The battery-powered area includes the RTC, SOSC, backup register, tamper detection and VBATT_R voltage drop detection and switch between VCC and VBATT. See <a href="#">section 11, Battery Backup Function</a> .
Register write protection	The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR_S and PRCR_NS). See <a href="#">section 12, Register Write Protection</a> .
Memory Protection Unit (MPU)	All bus masters have Memory Protection Units (MPUs). See <a href="#">section 15, Memory Protection Unit (MPU)</a> .

**Table 1.4 Event link**

Feature	Functional description
Event Link Controller (ELC)	The Event Link Controller (ELC) uses the event requests generated by various peripheral modules as source signals to connect them to different modules, allowing direct link between the modules without CPU intervention. See <a href="#">section 18, Event Link Controller (ELC)</a> .

**Table 1.5 Direct memory access**

Feature	Functional description
Data Transfer Controller (DTC)	A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request. See <a href="#">section 17, Data Transfer Controller (DTC)</a> .
DMA Controller (DMAC)	The 8-channel direct memory access controller (DMAC) that can transfer data without intervention from the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address. See <a href="#">section 16, DMA Controller (DMAC)</a> .

**Table 1.6 External bus interface**

Feature	Functional description
External buses	<ul style="list-style-type: none"> <li>OSPI area (EOBI): Connected to the OSPI (external device interface)</li> </ul>

**Table 1.7 Timers**

Feature	Functional description
General PWM Timer (GPT)	The General PWM Timer (GPT) is a 32-bit timer with GPT32 × 6 channels and a 16-bit timer with GPT16 × 4 channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. The GPT can also be used as a general-purpose timer. See <a href="#">section 21, General PWM Timer (GPT)</a> .
Port Output Enable for GPT (POEG)	The Port Output Enable (POEG) function can place the General PWM Timer (GPT) output pins in the output disable state See <a href="#">section 20, Port Output Enable for GPT (POEG)</a> .
Low Power Asynchronous General Purpose Timer (AGT)	The Low Power Asynchronous General Purpose Timer (AGT) is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGT register. See <a href="#">section 22, Low Power Asynchronous General Purpose Timer (AGT)</a> .
Ultra-Low-Power Timer (ULPT)	The Ultra-Low-Power Timer (ULPT) is a 32-bit timer which can be used for outputting pulses or counting external events. This 32-bit timer consists of reload registers and a down-counter. The reload registers and the down-counter are allocated to the same address and can be accessed through the ULPTCNT register. See <a href="#">section 23, Ultra-Low-Power Timer (ULPT)</a> .
Realtime Clock (RTC)	The realtime clock (RTC) has two counting modes, calendar count mode and binary count mode, that are used by switching register settings. For calendar count mode, the RTC has a 100-year calendar from 2000 to 2099 and automatically adjusts dates for leap years. For binary count mode, the RTC counts seconds and retains the information as a serial value. Binary count mode can be used for calendars other than the Gregorian (Western) calendar. See <a href="#">section 24, Realtime Clock (RTC)</a> .
Watchdog Timer (WDT)	The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt. See <a href="#">section 25, Watchdog Timer (WDT)</a> .
Independent Watchdog Timer (IWDT)	The Independent Watchdog Timer (IWDT) has a 14-bit down-counter, which resets the MCU by a reset output when the down-counter underflows. Alternatively, generation of an interrupt request when the counter underflows can be selected. This enables detection of a program runaway taking the refresh interval into account. The IWDT has two start modes: auto start mode, in which counting automatically starts after release from the reset state, and register start mode, in which counting is started by refreshing (writing to a specific register). See <a href="#">section 26, Independent Watchdog Timer (IWDT)</a> .

**Table 1.8 Communication interfaces**

Feature	Functional description
Serial Communications Interface (SCI)	<p>The Serial Communications Interface (SCI) × 6 channels have asynchronous and synchronous serial interfaces:</p> <ul style="list-style-type: none"> <li>• Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA))</li> <li>• 8-bit clock synchronous interface</li> <li>• Simple IIC (master-only)</li> <li>• Simple SPI</li> <li>• Smart card interface</li> <li>• Manchester interface</li> <li>• Simple LIN interface</li> </ul> <p>The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. All channels have FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator.</p> <p>The maximum rate supported on this MCU. Refer to the electrical characteristics for the actual rate.</p> <p>See <a href="#">section 30, Serial Communications Interface (SCI)</a>.</p>
I <sup>2</sup> C Bus interface (IIC)	<p>The I<sup>2</sup>C Bus interface (IIC) has 2 channels. The IIC module conforms with and provides a subset of the NXP I<sup>2</sup>C (Inter-Integrated Circuit) bus interface functions.</p> <p>See <a href="#">section 31, I<sup>2</sup>C Bus Interface (IIC)</a>.</p>
Serial Peripheral Interface (SPI)	<p>The Serial Peripheral Interface (SPI) provides high-speed full-duplex synchronous serial communications with multiple processors and peripheral devices.</p> <p>The maximum rate supported on this MCU. Refer to the electrical characteristics for the actual rate.</p> <p>See <a href="#">section 34, Serial Peripheral Interface (SPI)</a>.</p>
Control Area Network with Flexible Data-Rate Module (CANFD)	<p>The CAN with Flexible Data-Rate (CANFD) module can handle classical CAN frames and CANFD frames complied with ISO 11898-1 standard.</p> <p>The module supports 4 transmit buffers per channel and 16 receive buffers per channel.</p> <p>See <a href="#">section 32, CAN with Flexible Data-rate (CANFD)</a>.</p>
USB 2.0 Full-Speed module (USBFS)	<p>The USB 2.0 Full-Speed module (USBFS) can operate as a host controller or device controller. The module supports full-speed and low-speed (host controller only) transfer as defined in Universal Serial Bus Specification 2.0. The module has an internal USB transceiver and supports all of the transfer types defined in Universal Serial Bus Specification 2.0. The USB has buffer memory for data transfer, providing a maximum of 10 pipes. Pipes 1 to 9 can be assigned any endpoint number based on the peripheral devices used for communication or based on your system.</p> <p>See <a href="#">section 29, USB 2.0 Full-Speed Module (USBFS)</a>.</p>
Octal Serial Peripheral Interface (OSPI)	<p>The Octal Serial Peripheral Interface (OSPI) is a memory controller that supports EXpanded Serial Peripheral Interface (xSPI) (JEDEC Standard JESD251, JESD251-1 and JESD252) . The OSPI supports 1-bit, 2-bit, 4-bit and 8-bit protocols.</p> <p>JESD251 specifies two interface profiles where profile 1.0 is Octal SPI and profile 2.0 is HyperBus™ (HyperRAM™ and HyperFlash™ ).</p> <p>OSPI supports QSPI protocol.</p> <p>See <a href="#">section 35, Octal Serial Peripheral Interface (OSPI)</a>.</p>
Serial Sound Interface Enhanced (SSIE)	<p>The Serial Sound Interface Enhanced (SSIE) peripheral provides functionality to interface with digital audio devices for transmitting I<sup>2</sup>S/Monaural/TDM audio data over a serial bus. The SSIE supports an audio clock frequency of up to 50 MHz, and can be operated as a slave or master receiver, transmitter, or transceiver to suit various applications. The SSIE includes 32-stage FIFO buffers in the receiver and transmitter, and supports interrupts and DMA-driven data reception and transmission.</p> <p>See <a href="#">section 37, Serial Sound Interface Enhanced (SSIE)</a>.</p>
Ethernet Controller (ETHERC)	<p>One-channel Ethernet Controller (ETHERC) compliant with the Ethernet/IEEE802.3 Media Access Control (MAC) layer protocol. An ETHERC channel provides one channel of the MAC layer interface, connecting the MCU to the physical layer LSI (PHY-LSI) that allows transmission and reception of frames compliant with the Ethernet and IEEE802.3 standards. The ETHERC is connected to the Ethernet DMA Controller (EDMAC) so data can be transferred without using the CPU.</p> <p>See <a href="#">section 27, Ethernet MAC Controller (ETHERC)</a>.</p>

**Table 1.9 Analog**

Feature	Functional description
12-bit A/D Converter (ADC12)	A 12-bit successive approximation A/D Converter is provided. Up to 13 analog input channels are selectable. Temperature sensor output, and internal reference voltage and VBATT 1/3 voltage monitor are selectable for conversion. See <a href="#">section 42, 12-Bit A/D Converter (ADC12)</a> .
12-bit D/A Converter (DAC12)	A 12-bit D/A Converter (DAC12) is provided. See <a href="#">section 43, 12-Bit D/A Converter (DAC12)</a> .
Temperature Sensor (TSN)	The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC12 for conversion and can be further used by the end application. See <a href="#">section 44, Temperature Sensor (TSN)</a> .
High-Speed Analog Comparator (ACMPHS)	The High-Speed Analog Comparator (ACMPHS) can be used to compare an analog input voltage with a reference voltage and to provide a digital output based on the result of conversion. Both the analog input voltage and the reference voltage can be provided to the ACMPHS from internal sources (D/A converter output or internal reference voltage) and an external source. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion. See <a href="#">section 45, High-Speed Analog Comparator (ACMPHS)</a> .

**Table 1.10 Human machine interfaces**

Feature	Functional description
Capture Engine Unit (CEU)	The Capture Engine Unit (CEU) is a capture module that fetches image data externally input and transfers it to the memory. See <a href="#">section 50, Capture Engine Unit (CEU)</a> .

**Table 1.11 Data processing**

Feature	Functional description
Cyclic Redundancy Check (CRC) calculator	The Cyclic Redundancy Check (CRC) calculator generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generation polynomials are available. The snoop function allows monitoring reads from and writes to specific addresses. This function is useful in applications that require CRC code to be generated automatically in certain events, such as monitoring writes to the serial transmit buffer and reads from the serial receive buffer. See <a href="#">section 38, Cyclic Redundancy Check (CRC)</a> .
Data Operation Circuit (DOC)	The Data Operation Circuit (DOC) compares, adds, and subtracts 32-bits data. When a selected condition applies, 32-bit data is compared and an interrupt can be generated. See <a href="#">section 46, Data Operation Circuit (DOC)</a> .

**Table 1.12 Security**

Feature	Functional description
Security function	<ul style="list-style-type: none"> <li>● ARMv8-M TrustZone security</li> <li>● Privileged control</li> <li>● Device lifecycle management</li> <li>● Authentication Level (AL)</li> <li>● Key injection</li> <li>● Secure pin multiplexing</li> <li>● VBATT backup registers zeroization</li> <li>● Secure boot</li> <li>● Secure factory programming</li> </ul> See <a href="#">section 40, Security Features</a> .
Renesas Secure IP (RSIP-E51A)	<ul style="list-style-type: none"> <li>● 128-bit true random number generation circuit</li> <li>● 256-bit Hardware Unique Key (HUK)</li> <li>● 128-bit unique ID</li> </ul> See <a href="#">section 41, Renesas Secure IP (RSIP-E51A)</a> .

## 1.2 Block Diagram

Figure 1.1 shows a block diagram of the MCU superset. Some individual devices within the group have a subset of the features.

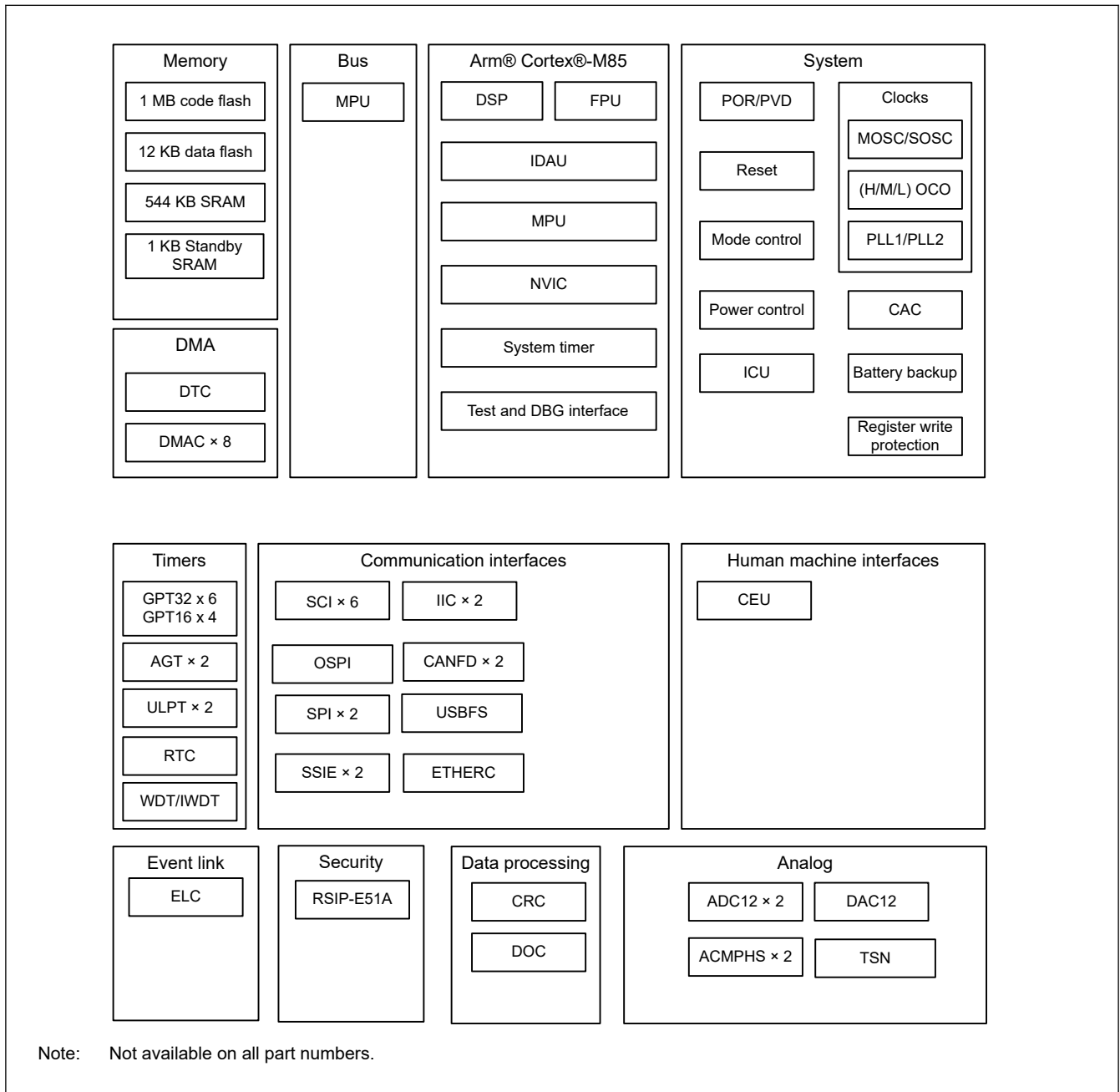


Figure 1.1 Block diagram

## 1.3 Part Numbering

Figure 1.2 shows the product part number information, including memory capacity and package type. Table 1.13 shows a list of products.

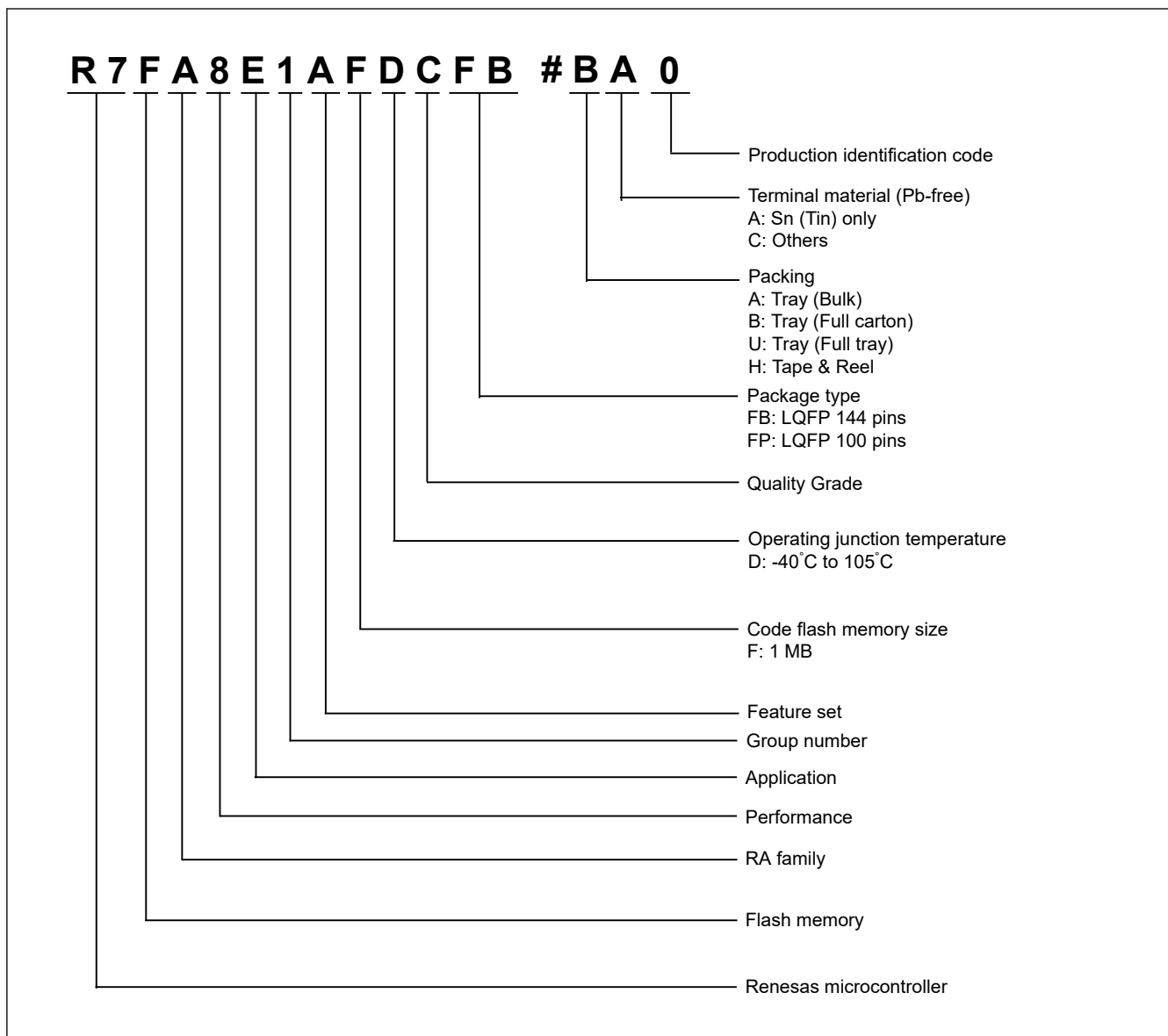


Figure 1.2 Part numbering scheme

Table 1.13 Product list

Product part number	Package code	Code flash	Data flash	SRAM	Operating junction temperature
R7FA8E1AFDCFB	PLQP0144KA-B	1 MB	12 KB	544 KB	-40 to +105°C
R7FA8E1AFDCFP	PLQP0100KP-A				

## 1.4 Function Comparison

Table 1.14 Function Comparison

Parts number		R7FA8E1AFDCFB	R7FA8E1AFDCFP
Pin count		144	100
Package		LQFP	
I/O Port		106	70
Code flash memory		1 MB	
Data flash memory		12 KB	
TCM		32 KB	
I/D Caches		32 KB	
SRAM	Parity	512 KB	
Standby SRAM		1 KB	
DMA	DTC	Yes	
	DMAC	8	
System	CPU clock	360 MHz (max.)	
	CPU clock sources	MOSC, SOSC, HOCO, MOCO, PLL1P	
	CAC	Yes	
	WDT/IWDT	Yes	
	Backup register	128 B	
Communication	SCI	6	
	IIC	2	
	SPI	2	
	CANFD	2	
	USBFS	Yes	
	OSPI	Yes	
	SSIE	2	1
	ETHERC	Yes	
Timers	GPT32*1	6	
	GPT16*1	4	
	AGT*1	2	
	ULPT*1	2	
	RTC	Yes	
Analog	ADC12	Unit 0: 8 Unit 1: 5	Unit 0: 6 Unit 1: 5
	DAC12	1	
	ACMPHS	2	
	TSN	Yes	
HMI	CEU	Yes	No
Data processing	CRC	Yes	
	DOC	Yes	
Event control	ELC	Yes	
Security		RSIP-E51A, Secure Debug, TrustZone, and Lifecycle management	

Note 1. Available pins depend on the Pin count. For details, see [section 1.7. Pin Lists](#).



## 1.5 Pin Functions

**Table 1.15 Pin functions (1 of 5)**

Function	Signal	I/O	Description
Power supply	VCC, VCC2	Input	Power supply pin. Connect it to the system power supply. Connect this pin to VSS by a 0.1- $\mu$ F capacitor. The capacitor should be placed close to the pin.
	VCC_DCDC	Input	Switching regulator power supply pin.
	VLO	I/O	Switching regulator pin.
	VCL	Input	Connect this pin to the VSS pin by the smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VBATT	Input	Battery Backup power pin
	VSS, VSS_DCDC	Input	Ground pin. Connect it to the system power supply (0 V).
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin.
	EXTAL	Input	
	XCIN	Input	Input/output pins for the sub-clock oscillator. Connect a crystal resonator between XCOU and XCIN.
	XCOU	Output	
	EXCIN	Input	External sub-clock input
	CLKOUT	Output	Clock output pin
Operating mode control	MD	Input	Pin for setting the operating mode. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
System control	RES	Input	Reset signal input pin. The MCU enters the reset state when this signal goes low.
CAC	CACREF	Input	Measurement reference clock input pin
On-chip emulator	TMS	Input	On-chip emulator or boundary scan pins
	TDI	Input	
	TCK	Input	
	TDO	Output	
	TCLK	Output	Output clock for synchronization with the trace data
	TDATA0 to TDATA3	Output	Trace data output
	SWO	Output	Serial wire trace output pin
	SWDIO	I/O	Serial wire debug data input/output pin
	SWCLK	Input	Serial wire clock pin
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQn	Input	Maskable interrupt request pins
	IRQn-DS	Input	Maskable interrupt request pins that can also be used in Deep Software Standby mode
GPT	GTETRG, GTETRGB, GTETRG, GTETRGD	Input	External trigger input pins
	GTIOCA, GTIOCB	I/O	Input capture, output compare, or PWM output pins
	GTADSM0, GTADSM1	Output	A/D conversion start request monitoring output pins

**Table 1.15 Pin functions (2 of 5)**

Function	Signal	I/O	Description
AGT	AGTEEn	Input	External event input enable signals
	AGTIO <sub>n</sub>	I/O	External event input and pulse output pins
	AGTO <sub>n</sub>	Output	Pulse output pins
	AGTOA <sub>n</sub>	Output	Output compare match A output pins
	AGTOB <sub>n</sub>	Output	Output compare match B output pins
ULPT	ULPTEEn	Input	External count control input
	ULPTEVIn	Input	External event input
	ULPTO <sub>n</sub>	Output	Pulse output
	ULPTOA <sub>n</sub>	Output	Output compare match A output
	ULPTOB <sub>n</sub>	Output	Output compare match B output
	ULPTEEn-DS	Input	External count control input that can also be used in Deep Software Standby mode1
	ULPTEVIn-DS	Input	External event input that can also be used in Deep Software Standby mode1
	ULPTO <sub>n</sub> -DS	Output	Pulse output that can also be used in Deep Software Standby mode1
	ULPTOA <sub>n</sub> -DS	Output	Output compare match A output that can also be used in Deep Software Standby mode1
	ULPTOB <sub>n</sub> -DS	Output	Output compare match B output that can also be used in Deep Software Standby mode1
RTC	RTCOU <sub>T</sub>	Output	Output pin for 1-Hz or 64-Hz clock
	RTCI <sub>Cn</sub>	Input	Time capture event input pins
SCI	SCK <sub>n</sub>	I/O	Input/output pins for the clock (clock synchronous mode)
	RXD <sub>n</sub>	Input	Input pins for received data (asynchronous mode/clock synchronous mode)
	TXD <sub>n</sub>	Output	Output pins for transmitted data (asynchronous mode/clock synchronous mode)
	CTS <sub>n</sub> _RTS <sub>n</sub>	I/O	Input/output pins for controlling the start of transmission and reception (asynchronous mode/clock synchronous mode), active-low.
	CTS <sub>n</sub>	Input	Input for the start of transmission.
	DE <sub>n</sub>	Output	Driver enable signal for RS-485
	SCL <sub>n</sub>	I/O	Input/output pins for the IIC clock (simple IIC mode)
	SDA <sub>n</sub>	I/O	Input/output pins for the IIC data (simple IIC mode)
	SCK <sub>n</sub>	I/O	Input/output pins for the clock (simple SPI mode)
	MISO <sub>n</sub>	I/O	Input/output pins for slave transmission of data (simple SPI mode)
	MOSI <sub>n</sub>	I/O	Input/output pins for master transmission of data (simple SPI mode)
	SS <sub>n</sub>	Input	Chip-select input pins (simple SPI mode), active-low
IIC	SCL <sub>n</sub>	I/O	Input/output pins for the clock
	SDA <sub>n</sub>	I/O	Input/output pins for data

Table 1.15 Pin functions (3 of 5)

Function	Signal	I/O	Description
SPI	RSPCKA, RSPCKB	I/O	Clock input/output pin
	MOSIA, MOSIB	I/O	Input or output pins for data output from the master
	MISOA, MISOB	I/O	Input or output pins for data output from the slave
	SSLA0, SSLB0	I/O	Input or output pin for slave selection
	SSLA1 to SSLA3, SSLB1 to SSLB3	Output	Output pins for slave selection
CANFD	CRXn	Input	Receive data
	CTXn	Output	Transmit data
USBFS	VCC_USB	Input	Power supply pin
	VSS_USB	Input	Ground pin
	USB_DP	I/O	D+ pin of the USB on-chip transceiver. Connect this pin to the D+ pin of the USB bus.
	USB_DM	I/O	D- pin of the USB on-chip transceiver. Connect this pin to the D- pin of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. Connect this pin to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB_EXICEN	Output	Low-power control signal for external power supply (OTG) chip
	USB_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip
	USB_OVRCURA, USB_OVRCURB	Input	Connect the external overcurrent detection signals to these pins. Connect the VBUS comparator signals to these pins when the OTG power supply chip is connected.
	USB_OVRCURA-DS, USB_OVRCURB-DS	Input	Overcurrent pins for USBFS that can also be used in Deep Software Standby mode1. Connect the external overcurrent detection signals to these pins. Connect the VBUS comparator signals to these pins when the OTG power supply chip is connected.
	USB_ID	Input	Connect the MicroAB connector ID input signal to this pin during operation in OTG mode
OSPI	OM_SCLK	Output	Clock output (OCTACK divided by 2)
	OM_SCLKN	Output	Inverted clock output (OCTACK divided by 2)
	OM_CS <sub>n</sub>	Output	Chip select signal for an OctaFlash device, active-low
	OM_DQS	I/O	Read data strobe/write data mask signal
	OM_SIO <sub>n</sub>	I/O	Data input/output
	OM_RESET	Output	Reset signal for both slave devices, active-low
	OM_ECSINT1	Input	Error Correction Status and Interrupt for slave1
	OM_RSTO1	Input	Slave reset status for slave1
	OM_WP1	Output	Write Protect for slave1, active-low
SSIE	SSIBCK0, SSIBCK1	I/O	SSIE serial bit clock pins
	SSILRCK0/SSIFS0, SSILRCK1/SSIFS1	I/O	LR clock/frame synchronization pins
	SSITXD0	Output	Serial data output pin
	SSIRXD0	Input	Serial data input pin
	SSIDATA1	I/O	Serial data input/output pins
	AUDIO_CLK	Input	External clock pin for audio (input oversampling clock)

Table 1.15 Pin functions (4 of 5)

Function	Signal	I/O	Description
ETHERC	REF50CK0	Input	50-MHz reference clock. This pin inputs reference signal for transmission/reception timing in RMII mode.
	RMII0_CRS_DV	Input	Indicates carrier detection signals and valid receive data on RMII0_RXD1 and RMII0_RXD0 in RMII mode
	RMII0_TXDn	Output	2-bit transmit data in RMII mode
	RMII0_RXDn	Input	2-bit receive data in RMII mode
	RMII0_TXD_EN	Output	Output pin for data transmit enable signal in RMII mode
	RMII0_RX_ER	Input	Indicates an error occurred during reception of data in RMII mode
	ET0_CRS	Input	Carrier detection/data reception enable signal
	ET0_RX_DV	Input	Indicates valid receive data on ET0_ERXD3 to ET0_ERXD0
	ET0_EXOUT	Output	General-purpose external output pin
	ET0_LINKSTA	Input	Input link status from the PHY-LSI
	ET0_ETXDn	Output	4 bits of MII transmit data
	ET0_ERXDn	Input	4 bits of MII receive data
	ET0_TX_EN	Output	Transmit enable signal. Functions as signal indicating that transmit data is ready on ET0_ETXD3 to ET0_ETXD0.
	ET0_TX_ER	Output	Transmit error pin. Functions as signal notifying the PHY_LSI of an error during transmission.
	ET0_RX_ER	Output	Receive error pin. Functions as signal to recognize an error during reception.
	ET0_TX_CLK	Input	Transmit clock pin. This pin inputs reference signal for output timing from ET0_TX_EN, ET0_ETXD3 to ET0_ETXD0, and ET0_TX_ER.
	ET0_RX_CLK	Input	Receive clock pin. This pin inputs reference signal for input timing to ET0_RX_DV, ET0_ERXD3 to ET0_ERXD0, and ET0_RX_ER.
	ET0_COL	Input	Input collision detection signal
	ET0_WOL	Output	Receive Magic packets
	ET0_MDC	Output	Output reference clock signal for information transfer through ET0_MDIO
ET0_MDIO	I/O	Input or output bidirectional signal for exchange of management data with PHY-LSI	
Analog power supply	AVCC0	Input	Analog voltage supply pin. This is used as the analog power supply for the respective modules.
	AVSS0	Input	Analog ground pin. This is used as the analog ground for the respective modules. Supply this pin with the same voltage as the VSS pin.
	VREFH	Input	Analog reference voltage supply pin for the ADC12 (unit 1) and D/A Converter. Connect this pin to AVCC0 when not using the ADC12 (unit 1) and D/A Converter.
	VREFL	Input	Analog reference ground pin for the ADC12 and D/A Converter. Connect this pin to AVSS0 when not using the ADC12 (unit 1) and D/A Converter.
	VREFH0	Input	Analog reference voltage supply pin for the ADC12 (unit 0). Connect this pin to AVCC0 when not using the ADC12 (unit 0).
	VREFL0	Input	Analog reference ground pin for the ADC12. Connect this pin to AVSS0 when not using the ADC12 (unit 0).

**Table 1.15 Pin functions (5 of 5)**

Function	Signal	I/O	Description
ADC12	ANmn	Input	Input pins for the analog signals to be processed by the A/D converter. (m: ADC unit number, n: pin number)
	ADTRGm	Input	Input pins for the external trigger signals that start the A/D conversion, active-low.
DAC12	DA0	Output	Output pin for the analog signals processed by the D/A converter.
ACMPHS	VCOUT	Output	Comparator output pin
	IVREFn	Input	Reference voltage input pins for comparator
	IVCMPn	Input	Analog voltage input pins for comparator
I/O ports	Pmn	I/O	General-purpose input/output pins (m: port number, n: pin number)
	P200	Input	General-purpose input pin
CEU	VIO_D7 to VIO_D0	Input	CEU data bus pins
	VIO_CLK	Input	CEU clock pin
	VIO_VD	Input	CEU vertical sync pin
	VIO_HD	Input	CEU horizontal sync pin

## 1.6 Pin Assignments

The following figures show the pin assignments from the top view.

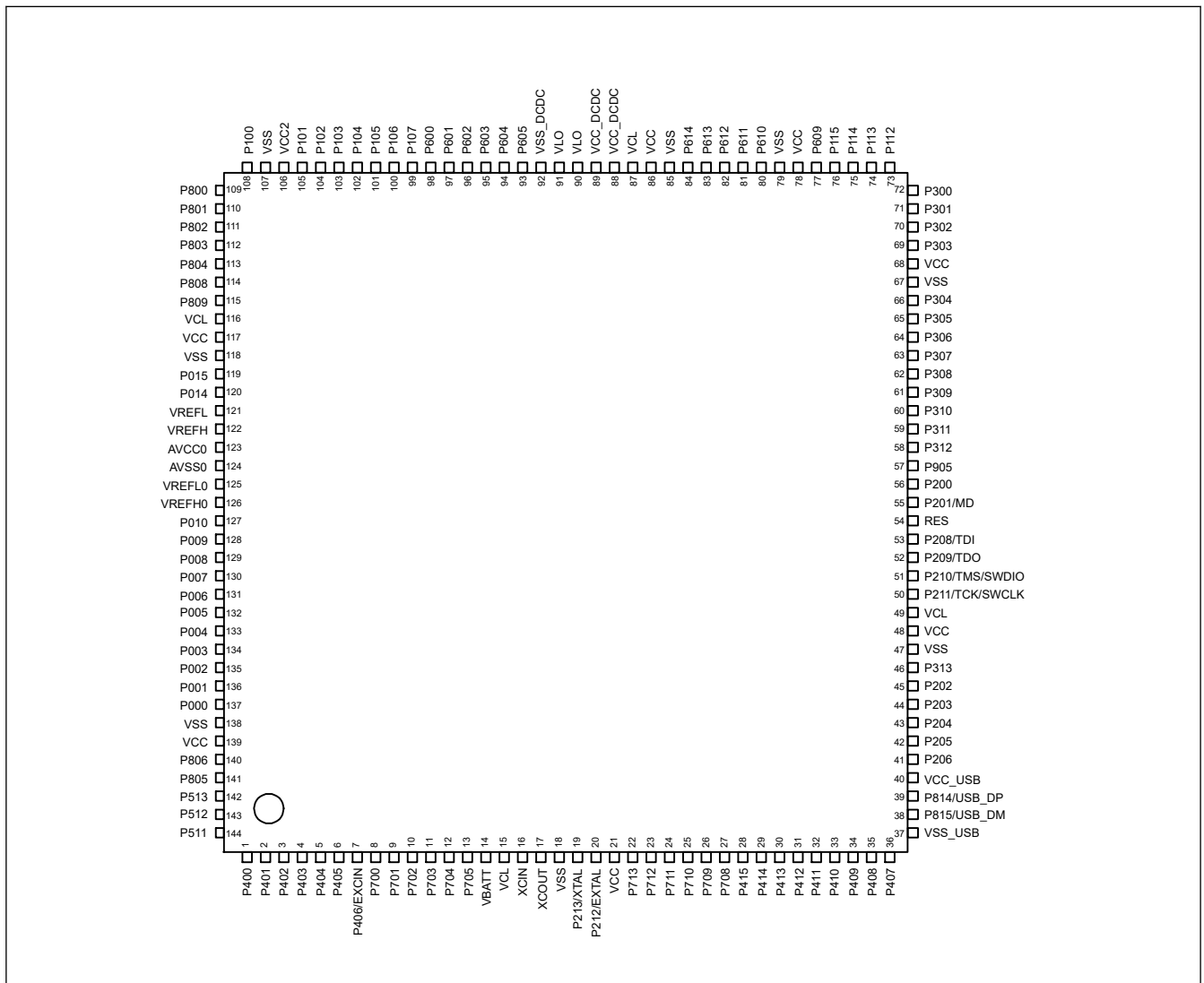


Figure 1.3 Pin assignment for LQFP 144-pin

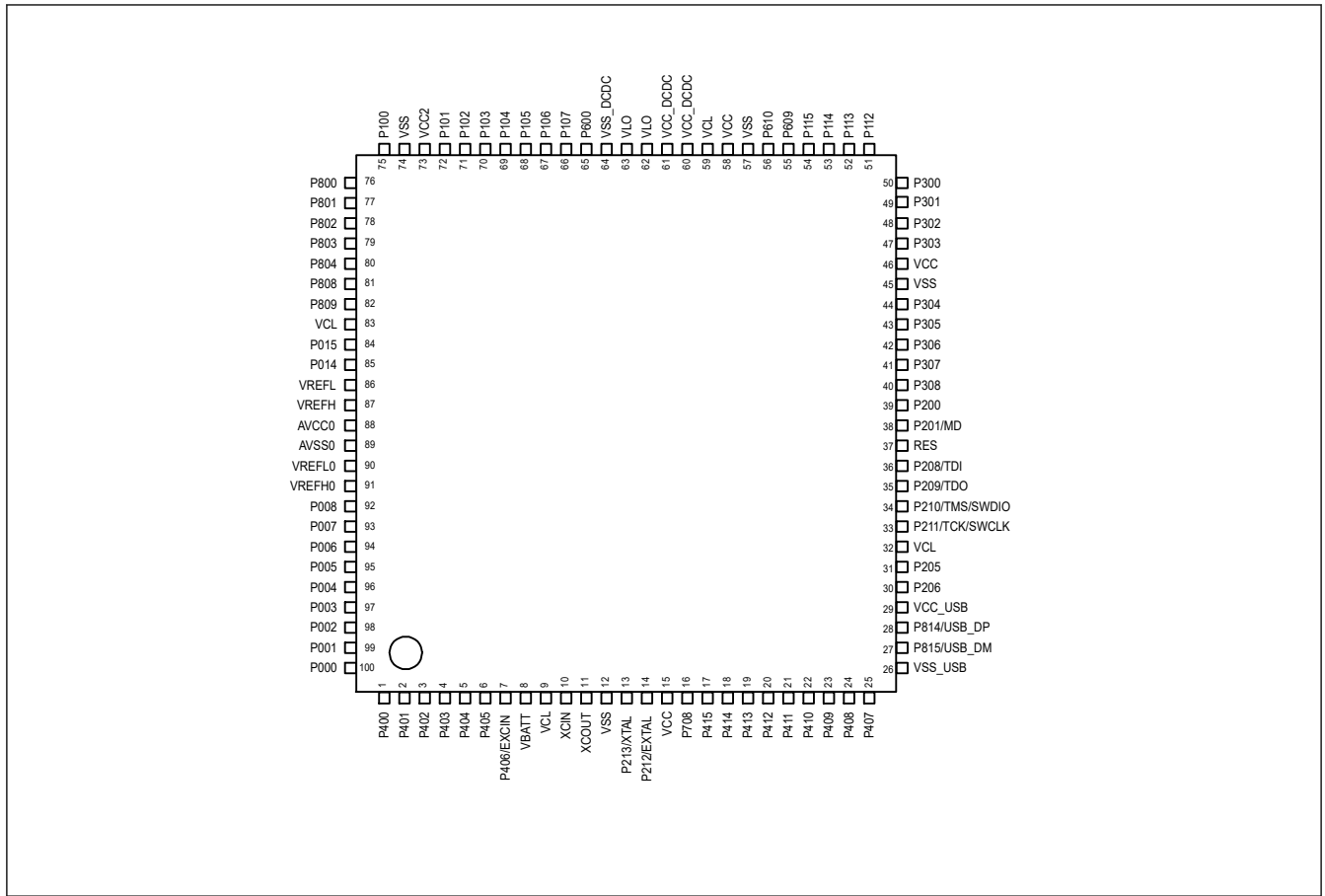


Figure 1.4 Pin assignment for LQFP 100-pin

1.7 Pin Lists

Table 1.16 Pin list (1 of 4)

LGFP144	LGFP100	Power, System, Clock, Debug, CAC	I/O ports	Ex. Interrupt	SCI/IIC/SPI/CANFD/USBFS/OSPI/SSIE/EHTERC(MII,RMII)	GPT/AGT/ULPT/RTC	ADC12/DAC12/ACMPHS	CEU
1	1	-	P400	IRQ0	TXD1_A/MOSI1_A/SDA1_A/AUDIO_CLK/ET0_WOL/ET0_WOL	AGTIO1	ADTRG1	VIO_D0
2	2	-	P401	IRQ5-DS	RXD1_A/MISO1_A/SCL1_A/CTX0/ET0_MDC/ET0_MDC	GTETRGA	-	VIO_D1
3	3	CACREF	P402	IRQ4-DS	SCK1_A/DE1/CRX0/AUDIO_CLK/ET0_MDIO/ET0_MDIO	RTIC0	-	-
4	4	-	P403	IRQ14-DS	CTS_RTS4_A/SS4_A/DE1/SSIBCK0_A/ET0_LINKSTA/ET0_LINKSTA	GTIOC3A/RTIC1	-	-
5	5	-	P404	IRQ15-DS	CTS1_A/SSILRCK0/SSIFS0_A/ET0_EXOUT/ET0_EXOUT	GTIOC3B/RTIC2	-	VIO_D3
6	6	-	P405	-	SCK2_B/DE2/SSITXD0_A/ET0_TX_EN/RMII0_TXD_EN_B	GTIOC1A/AGTIO1	-	VIO_D2
7	7	EXCIN	P406	-	TXD2_B/MOSI2_B/SDA2_B/SSLA3_C/SSIRXD0_A/ET0_RX_ER/RMII0_TXD1_B	GTIOC1B	-	VIO_D3
8	-	-	P700	-	RXD2_B/MISO2_B/SCL2_B/MISOA_C/SSIDATA1_B/ET0_ETXD1/RMII0_TXD0_B	GTIOC5A	-	VIO_D4
9	-	-	P701	-	CTS_RTS2_B/SS2_B/DE2/MOSIA_C/SSILRCK1/SSIFS1_B/ET0_ETXD0/REF50CK0_B	GTIOC5B/ULPT01	-	VIO_D5
10	-	-	P702	-	CTS2_B/RSPCKA_C/SSIBCK1_B/ET0_ERXD1/RMII0_RXD0_B	ULPT00	-	VIO_D6
11	-	-	P703	-	SSLA0_C/ET0_ERXD0/RMII0_RXD1_B	AGT01	VCOUT	VIO_D7
12	-	-	P704	-	SSLA1_C/CTX0/ET0_RX_CLK/RMII0_RX_ER_B	GTADSM0/AGT00	-	-
13	-	-	P705	-	CTS1_B/SSLA2_C/CRX0/ET0_CRS/RMII0_CRS_DV_B	GTADSM1/AGTIO0	-	-
14	8	VBATT	-	-	-	-	-	-
15	9	VCL	-	-	-	-	-	-
16	10	XCIN	-	-	-	-	-	-
17	11	XCOUT	-	-	-	-	-	-
18	12	VSS	-	-	-	-	-	-
19	13	XTAL	P213	IRQ2	TXD1_C/MOSI1_C/SDA1_C	GTETRGC/GTIOC0A/ULPTEE0	ADTRG1	-
20	14	EXTAL	P212	IRQ3	RXD1_C/MISO1_C/SCL1_C	GTETRGD/GTIOC0B/AGTEE1	-	-
21	15	VCC	-	-	-	-	-	-
22	-	-	P713	-	-	GTIOC2A/AGTOA0	-	-
23	-	-	P712	-	-	GTIOC2B/AGTOB0	-	-
24	-	-	P711	-	-	AGTEE0	-	-
25	-	-	P710	-	CTS4_B	-	-	VIO_VD
26	-	-	P709	IRQ10	CTS_RTS4_B/SS4_B/DE4	-	-	VIO_HD
27	16	CACREF	P708	IRQ11	SCK4_B/DE4/SSLB3_B/AUDIO_CLK	-	-	VIO_CLK
28	17	-	P415	IRQ8	TXD4_B/MOSI4_B/SDA4_B/SSLB2_B/CTX1	GTADSM0/GTIOC0A	-	-
29	18	-	P414	IRQ9	RXD4_B/MISO4_B/SCL4_B/SSLB1_B/CRX1	GTADSM1/GTIOC0B	-	-
30	19	-	P413	-	SSLB0_B	ULPTEE1	-	-
31	20	-	P412	-	CTS3_A/RSPCKB_B/USB_EXICEN	AGTEE1	-	-
32	21	-	P411	IRQ4	CTS_RTS3_A/SS3_A/DE3/MOSIB_B/USB_ID	AGTOA1	-	-
33	22	-	P410	IRQ5	SCK3_A/DE3/SCL0_A/MISOB_B/USB_OVRCURB-DS	AGTOB1	-	-
34	23	-	P409	IRQ6	TXD3_A/MOSI3_A/SDA3_A/SDA0_A/USB_OVRCURA-DS	ULPTOA0	-	-
35	24	-	P408	IRQ7	CTS4_A/RXD3_A/MISO3_A/SCL3_A/SCL0_B/USB_VBUSEN	GTIOC10A/ULPTOB0	-	-
36	25	-	P407	-	CTS_RTS4_A/SS4_A/DE4/SDA0_B/SSLA3_A/USB_VBUS	GTIOC10B/AGTIO0/RTCOUT	ADTRG0	-
37	26	VSS_USB	-	-	-	-	-	-
38	27	-	P815	-	CTX0/USB_DM	-	-	-
39	28	-	P814	-	CRX0/USB_DP	-	-	-
40	29	VCC_USB	-	-	-	-	-	-
41	30	-	P206	IRQ0-DS	RXD4_A/MISO4_A/SCL4_A/SDA1_B/SSLA2_A/USB_VBUSEN/SSIDATA1_A	-	-	-



Table 1.16 Pin list (2 of 4)

LOFP144	LOFP100	Power, System, Clock, Debug, CAC	I/O ports	Ex. Interrupt	SCI/IIC/SPI/CANFD/USBFS/OSPI/SSIE/EHTERC(MII,RMII)	GPT/AGT/ULPT/RTC	ADC12/DAC12/ACMPHS	CEU
42	31	CLKOUT	P205	IRQ1-DS	TXD4_A/MOSI4_A/SDA4_A/SCL1_B/SSLA1_A/ USB_OVRCURA/SSILRCK1/SSIFS1_A	GTIOC4A/AGTO1	-	-
43	-	CACREF	P204	-	SCK4_A/DE4/SSLA0_A/USB_OVRCURB/SSIBCK1_A	GTIOC4B/AGTIO1	-	-
44	-	-	P203	IRQ2-DS	RSPCKA_A/CTX0	GTIOC5A/ULPTOA1	-	-
45	-	-	P202	IRQ3-DS	MOSIA_A/CRX0	GTIOC5B/ULPTOB1	-	-
46	-	-	P313	-	CTS3_C/MISOA_A	-	-	-
47	-	VSS	-	-	-	-	-	-
48	-	VCC	-	-	-	-	-	-
49	32	VCL	-	-	-	-	-	-
50	33	TCK/SWCLK	P211	-	SCK9_B/DE9	GTIOC0A	-	-
51	34	TMS/SWDIO	P210	-	CTS_RTS9_B/SS9_B/DE9	GTIOC0B	-	-
52	35	TDO/SWO/CLKOUT	P209	-	TXD9_B/MOSI9_B/SDA9_B/CTX1	GTIOC1A	-	-
53	36	TDI	P208	IRQ3	RXD9_B/MISO9_B/SCL9_B/CRX1	GTIOC1B	VCOUT	-
54	37	RES	-	-	-	-	-	-
55	38	MD	P201	-	-	-	-	-
56	39	-	P200	NMI	-	-	-	-
57	-	-	P905	IRQ8	CTS3_B	-	-	-
58	-	-	P312	-	CTS_RTS3_B/SS3_B/DE3/CTX0/ET0_TX_CLK	GTADSM0/AGTOA1	-	-
59	-	-	P311	-	SCK3_B/DE3/CRX0/ET0_TX_ER	GTADSM1/AGTOB1	-	-
60	-	-	P310	-	TXD3_B/MOSI3_B/SDA3_B/ET0_ETXD2	AGTEE1	-	-
61	-	-	P309	-	RXD3_B/MISO3_B/SCL3_B/ET0_ETXD3	-	-	-
62	40	TCLK	P308	-	CTS9_B/ET0_MDC/ET0_MDC	ULPTOB1	-	-
63	41	TDATA0	P307	-	ET0_MDIO/ET0_MDIO	ULPTOA1	-	-
64	42	TDATA1	P306	-	ET0_TX_EN/RMII0_TXD_EN_A	ULPTEV11	-	-
65	43	TDATA2	P305	IRQ8	ET0_RX_ER/RMII0_TXD1_A	ULPTEE1	-	-
66	44	TDATA3	P304	IRQ9	ET0_ETXD1/RMII0_TXD0_A	ULPTO1	-	-
67	45	VSS	-	-	-	-	-	-
68	46	VCC	-	-	-	-	-	-
69	47	-	P303	-	ET0_ETXD0/REF50CK0_A	-	-	-
70	48	-	P302	IRQ5	ET0_ERXD1/RMII0_RXD0_A	GTIOC4A/ULPTO0-DS	-	-
71	49	-	P301	IRQ6	ET0_ERXD0/RMII0_RXD1_A	GTIOC4B/AGTIO0/ ULPTEE0-DS	-	-
72	50	-	P300	IRQ4	SCK0_A/DE0/SSLA3_B/ET0_RX_CLK/RMII0_RX_ER_A	GTIOC3A/ULPTEVIO-DS	-	-
73	51	-	P112	-	TXD0_A/MOSI0_A/SDA0_A/SSLA2_B/SSIBCK0_B/ ET0_CRS/RMII0_CRS_DV_A	GTIOC3B/ULPTOB0-DS	-	-
74	52	-	P113	-	RXD0_A/MISO0_A/SCL0_A/SSLA1_B/SSILRCK0/ SSIFS0_B/ET0_EXOUT/ET0_EXOUT	GTIOC2A/ULPTOA0-DS	-	-
75	53	-	P114	-	CTS0_RTS0_A/SS0_A/DE0/SSLA0_B/SSIRXD0_B/ ET0_LINKSTA/ET0_LINKSTA	GTIOC2B	-	-
76	54	-	P115	-	CTS0_A/MOSIA_B/SSITXD0_B/ET0_WOL/ET0_WOL	GTIOC5A	-	-
77	55	-	P609	-	TXD0_C/MOSI0_C/SDA0_C/MISOA_B/CTX1/ET0_RX_DV	GTIOC5B/ULPTOA1-DS	-	-
78	-	VCC	-	-	-	-	-	-
79	-	VSS	-	-	-	-	-	-
80	56	-	P610	-	RXD0_C/MISO0_C/SCL0_C/RSPCKA_B/CRX1/ET0_COL	GTIOC4A/ULPTOB1-DS	-	-
81	-	CLKOUT/CACREF	P611	-	SCK0_C/DE0/MOSIA_B/ET0_ERXD2	GTIOC4B	-	-
82	-	-	P612	-	CTS_RTS0_C/SS0_C/DE0/SSLA0_B/ET0_ERXD3	-	-	-
83	-	-	P613	-	CTS0_C	GTETRG/AGTO1	-	-
84	-	-	P614	-	-	GTETRGB/AGTO0	-	-
85	57	VSS	-	-	-	-	-	-
86	58	VCC	-	-	-	-	-	-
87	59	VCL	-	-	-	-	-	-

**Table 1.16 Pin list (3 of 4)**

LOFP144	LOFP100	Power, System, Clock, Debug, CAC	I/O ports	Ex. Interrupt	SCI/IIC/SPI/CANFD/USBFS/OSPI/SSIE/EHTERC(MII,RMI)	GPT/AGT/ULPT/RTC	ADC12/DAC12/ACMPHS	CEU
88	60	VCC_DCDC	-	-	-	-	-	-
89	61	VCC_DCDC	-	-	-	-	-	-
90	62	VLO	-	-	-	-	-	-
91	63	VLO	-	-	-	-	-	-
92	64	VSS_DCDC	-	-	-	-	-	-
93	-	-	P605	-	CTS0_B	-	-	-
94	-	-	P604	-	CTS_RTS0_B/SS0_B/DE0	-	-	-
95	-	-	P603	-	TXD0_B/MOSI0_B/SDA0_B	ULPT00	-	-
96	-	-	P602	-	RXD0_B/MISO0_B/SCL0_B	ULPTEE0	-	-
97	-	-	P601	-	SCK0_B/DE0/OM_WP1	ULPTEVI0/RTCOUT	-	-
98	65	CACREF	P600	-	OM_RST01	ULPTEVI1-DS	-	-
99	66	-	P107	-	OM_CS0	AGTOA0	-	-
100	67	-	P106	-	SSLB3_A/OM_RESET	AGTOB0/ULPTEE1-DS	-	-
101	68	-	P105	IRQ0	SSLB2_A/OM_ECSINT1	GTIOC1A/ULPT01-DS	-	-
102	69	-	P104	IRQ1	CTS9_A/SSLB1_A/OM_CS1	GTETRGB/GTIOC1B	-	-
103	70	-	P103	-	CTS9_RTS9_A/SS9_A/DE9/SSLB0_A/CTX0/OM_SIO2	GTIOC2A	-	-
104	71	-	P102	-	TXD9_A/MOSI9_A/SDA9_A/RSPCKB_A/CRX0/OM_SIO4	GTIOC2B/AGTO0	ADTRG0	-
105	72	-	P101	IRQ1	RXD9_A/MISO9_A/SCL9_A/MOSIB_A/OM_SIO3	GTETRGB/AGTEE0	-	-
106	73	VCC2	-	-	-	-	-	-
107	74	VSS	-	-	-	-	-	-
108	75	-	P100	IRQ2	SCK9_A/DE9/MISOB_A/OM_SIO0	GTETRGA/AGTIO0	-	-
109	76	-	P800	IRQ11	CTS2_A/OM_SIO5	GTIOC11A/AGTOA0	-	-
110	77	-	P801	IRQ12	TXD2_A/MOSI2_A/SDA2_A/OM_DQS	GTIOC11B/AGTOB0	-	-
111	78	-	P802	-	RXD2_A/MISO2_A/SCL2_A/OM_SIO6	GTIOC12A	-	-
112	79	-	P803	-	SCK2_A/DE2/OM_SIO1	GTETRGC/GTIOC12B	-	-
113	80	-	P804	IRQ14	CTS_RTS2_A/SS2_A/DE2/OM_SIO7	GTETRGD/GTIOC13A	-	-
114	81	-	P808	IRQ15	OM_SCLK	GTIOC13B	-	-
115	82	-	P809	-	OM_SCLKN	-	-	-
116	83	VCL	-	-	-	-	-	-
117	-	VCC	-	-	-	-	-	-
118	-	VSS	-	-	-	-	-	-
119	84	-	P015	IRQ13	-	-	AN105	-
120	85	-	P014	-	-	-	AN007/DA0	-
121	86	VREFL	-	-	-	-	-	-
122	87	VREFH	-	-	-	-	-	-
123	88	AVCC0	-	-	-	-	-	-
124	89	AVSS0	-	-	-	-	-	-
125	90	VREFLO	-	-	-	-	-	-
126	91	VREFH0	-	-	-	-	-	-
127	-	-	P010	IRQ14	-	-	AN005/IVCMP0	-
128	-	-	P009	IRQ13-DS	-	-	AN006	-
129	92	-	P008	IRQ12-DS	-	-	AN008	-
130	93	-	P007	-	-	-	AN004	-
131	94	-	P006	IRQ11-DS	-	-	AN002/IVCMP3	-
132	95	-	P005	IRQ10-DS	-	-	AN001	-
133	96	-	P004	IRQ9-DS	-	-	AN000/IVCMP2	-
134	97	-	P003	-	-	-	AN104/IVREF1	-
135	98	-	P002	IRQ8-DS	-	-	AN102/IVCMP3	-
136	99	-	P001	IRQ7-DS	-	-	AN101/IVREF0	-

**Table 1.16 Pin list (4 of 4)**

LOFP144	LOFP100	Power, System, Clock, Debug, CAC	I/O ports	Ex. Interrupt	SCI/IIC/SPI/CANFD/USBFS/OSP/SSIE/EHTERC(MII,RMII)	GPT/AGT/ULPT/RTC	ADC12/DAC12/ACMPHS	CEU
137	100	-	P000	IRQ6-DS	-	-	AN100/IVCMP2	-
138	-	VSS	-	-	-	-	-	-
139	-	VCC	-	-	-	-	-	-
140	-	-	P806	IRQ0	-	-	-	-
141	-	-	P805	-	-	-	-	-
142	-	-	P513	-	-	-	IVCMP0	VIO_FLD
143	-	-	P512	IRQ14	SCL1_A/CTX1	GTIOC0A	-	-
144	-	-	P511	IRQ15	SDA1_A/CRX1	GTIOC0B	-	-

Note: Several pin names have the added suffix of \_A, \_B, and \_C. These suffixes have special conditions for electrical characteristics. See [section 52, Electrical Characteristics](#) for detail.

## 2. CPU

The MCU is based on the Arm<sup>®</sup> Cortex<sup>®</sup>-M85 Processor.

### 2.1 Overview

#### 2.1.1 CPU

- Arm<sup>®</sup> Cortex<sup>®</sup>-M85 Processor
  - Revision: r0p2-00rel0
  - ARMv8.1-M architecture profile
  - Armv8-M Security Extension
- Extension Processing Unit (EPU)
  - Floating Point Unit (FPU) compliant with the ANSI/IEEE Std 754-2008 Scalar half, single, and double-precision floating-point operation
  - M-profile Vector Extension (MVE)  
Integer, half-precision, and single-precision floating-point MVE (MVE-F)
- Security Attribution Unit (SAU): 8 regions
- Implementation Defined Attribution Unit (IDAU)
- Memory Protection Unit (MPU)
  - Protected Memory System Architecture (PMSAv8)
  - Secure MPU (MPU\_S): 8 regions
  - Non-secure MPU (MPU\_NS): 8 regions
- System Timer (SysTick)
  - Reference clock:
    - CPUCLK
    - SYSTICKCLK
  - Two system timers are implemented:
    - Secure instance (SysTick\_S)
    - Non-secure instance (SysTick\_NS)
- Low power modes
  - CPU Sleep mode
  - CPU Deep Sleep mode
- Cache
  - Instruction cache: 16 KB with ECC
  - Data cache: 16 KB with ECC
- Tightly Coupled Memory (TCM)
  - ITCM: 16 KB with ECC (8 KB × 2 block)
  - DTCM: 16 KB with ECC (8 KB × 2 block)
- Nested Vectored Interrupt Controller (NVIC): 96 IRQs

See [section 2.14. References](#) [1], [2] for the detail.

## 2.1.2 Debug

### CPU Debug Components

- Arm® CoreSight™ ETM-M85
  - Revision: r0p2-00rel0
  - ARM ETM Architecture version 4.5
- Instrumentation Trace Macrocell (ITM)
- Data Watchpoint and Trace unit (DWT)
  - 8 Data Watchpoint and Trace (DWT) comparators, up to 2 data value comparison
- Breakpoint Unit (BPU)
  - 8 instruction comparators
- Cortex®-M85 Trace Port Interface Unit (TPIU)
  - 4 bits TPIU Formatter output
  - Serial Wire Output (SWO)

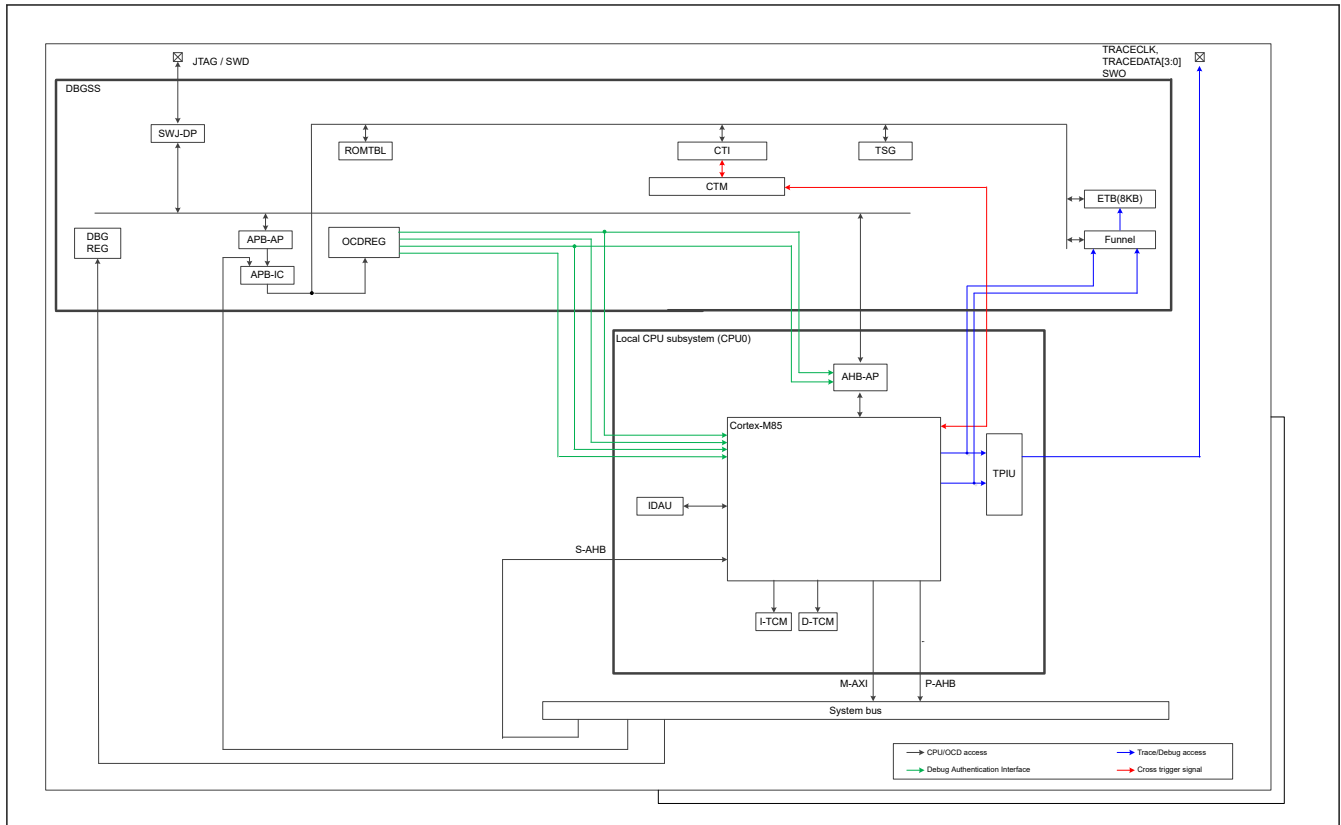
### Common Debug Components

- Debug Access Port (DAP)
  - CoreSight SoC-400 Debug Access Port
  - JTAG Debug Port and Serial Wire Debug Port (SWJ-DP)  
Serial Wire protocol version 2 (target instance is fixed value: 0b0000)
- Timestamp Generator (TSG)
  - CoreSight SoC-400 Timestamp Generator
  - Global Timestamp for ETM and ITM
  - Driven by DCLK
- Cross Trigger Interface (CTI)
  - CoreSight SoC-400 CTI (CTI)
  - Processor CTI (CTI0)
- Embedded Trace Buffer (ETB)
  - CoreSight™ Trace Memory Controller with ETB configuration
  - Buffer size: 8 KB
- Debug Register Module (DBGREG)
- On-Chip Debug Register Module (OCDREG)
- Authentication method through JTAG/SWD: Challenge response

See [section 2.14. References \[1\], \[2\]](#) for the detail.

## 2.2 Block Diagram

[Figure 2.1](#) shows the block diagram of CPU and CoreSight Debug subsystem (DBGSS).



**Figure 2.1** Block diagram of CPU and CoreSight Debug subsystem (DBGSS)

Note: See section 2.14. References [2] for detail of Cortex-M85 block diagram. Cortex-M85 block diagram described in this document is just for illustrative purposes only.

### 2.3 Implementation Options

Table 2.1 shows the implementation options of the MCU.

**Table 2.1** Implementation options (1 of 2)

Option	Implementation
SAU	8 regions
IDAU	Included
MPU	MPU_S: 8 regions MPU_NS: 8 regions
MVE	Integer and half and single-precision floating-point MVE
FPU	Scalar half, single, and double-precision floating-point
Interrupts	96
Priority bits	4 bits (16 levels)
Sleep mode power-saving	CPU Sleep mode, CPU Deep Sleep mode
TCM	<ul style="list-style-type: none"> <li>ITCM : 16 KB</li> <li>DTCM : 16 KB</li> <li>ECC enable (determined by OFS1(_SEC).INITECCEN)</li> <li>ITCMCR.EN = 1 and DTCMCR.EN = 1, cannot be rewrite</li> </ul>
CACHE	<ul style="list-style-type: none"> <li>I-Cache: 16 KB</li> <li>D-Cache: 16 KB</li> <li>ECC enable (determined by OFS1(_SEC).INITECCEN)</li> <li>Automatic invalidation is enabled</li> </ul> During debug, it is controlled by CACHEDBGCR L1RSTDIS

**Table 2.1 Implementation options (2 of 2)**

Option	Implementation
P-AHB	<ul style="list-style-type: none"> <li>PAHBCR.SZ is 0b011 (256 MB)</li> <li>PAHBCR.EN is always 1</li> </ul>
SysTick	Included two system timers, SysTick_S and SysTick_NS External reference clock with frequency = SYSTICKCLK SYST_CALIB = 0x0000270F NOREF, bit[31] = 0: Reference clock is implemented SKEW, bit[30] = 0: TENMS calibration value is exact Bit[29:24] = 0x00: Reserved TENMS, bit[23:0] = 0x00270F: $(0x270F + 1) \times 1 \mu s = 10 \text{ ms}$
PACBTI	Included and Enabled
Software reset request output	AIRCR.SYSRESETREQ bit causes the Software reset
DWT	8 comparators
BPU	8 comparators
CTI	Included, CTI and CTI0
ITM	Included
ETM	Included
TPIU	Included <ul style="list-style-type: none"> <li>4-bit TPIU formatter output</li> <li>Serial Wire Output (SWO)</li> </ul>

**Table 2.2 DBGSS implementation options**

Option	Implementation
TSG	Included Driven by clock with frequency = ICLK/2
ETB	8 KB
DAP	SWJ-DP AHB-AP : AP0 APB-AP : AP1

## 2.4 Trace Interface

A Trace Port Interface Unit (TPIU) and Serial Wire Output (SWO) provide trace output. [Table 2.3](#) shows the MCU pins for the function. These pins are multiplexed with other functions.

**Table 2.3 Trace function pins**

Name	I/O	Width	Function	When not in use
TCLK	Out	1 bit	Trace clock	Open
TDATA0	Out	1 bit	Trace data output	Open
TDATA1	Out	1 bit	Trace data output	Open
TDATA2	Out	1 bit	Trace data output	Open
TDATA3	Out	1 bit	Trace data output	Open
SWO	Out	1 bit	Serial Wire Output	Open

## 2.5 JTAG/SWD Interface

[Table 2.4](#) shows JTAG/SWD pins.

**Table 2.4 JTAG/SWD pins (1 of 2)**

Name	I/O	P/N	Width	Function	When not in use
TCK/SWCLK	In	Pos.	1 bit	JTAG Clock pin Serial Wire Data Clock pin	Pull-up

**Table 2.4 JTAG/SWD pins (2 of 2)**

Name	I/O	P/N	Width	Function	When not in use
TMS/SWDIO	In-Out	Neg.	1 bit	JTAG TMS pin Serial Wire Data In-Out pin	Pull-up
TDI	In	Pos.	1 bit	JTAG TDI pin	Pull-up
TDO	Out	Neg.	1 bit	JTAG TDO pin	Open

## 2.6 Initial Vector Table Base Address

In the MCU, the secure vector table base address is determined by the operating mode as shown in [Table 2.5](#), and the non-secure vector table base address is fixed at 0x0000\_0000.

**Table 2.5 Initial vector table base address**

Operating mode	CPU INITSVTOR
Single-chip mode	0x0200_0000

## 2.7 Debug Function

The MCU provides ability to debug software by either external debug or self-hosted debug.

The external debug function can be enabled by JTAG/SWD. See [section 2.7.1. External Debugger Connectivity and Authentication Through JTAG/SWD](#).

Self-hosted debug function can be enabled by software programming. See [section 2.7.2. Debug Protection Mechanism by Software](#) for details.

When debug function is enabled, operation of system can be affected. See [section 2.7.3. Effect of Debug Function](#) for details.

### 2.7.1 External Debugger Connectivity and Authentication Through JTAG/SWD

When debug function is enabled by performing authentication through JTAG/SWD, the three debug levels that correspond to the three Authentication Level (AL) are AL0, AL1, and AL2.

- AL0: No debug functions\*<sup>1</sup> are available.
- AL1: Only non-secure debug functions\*<sup>1</sup> is enabled, and debugger can access only defined Non-secure debug accessible regions. It is defined as Non-secure debug in [section 2.14. References \[1\]](#).
- AL2: Non-secure and Secure debug functions\*<sup>1</sup> are enabled and accessible from the debugger. It is defined as Secure debug in [section 2.14. References \[1\]](#).

Authentication Level is determined as follows:

- When Device Life Cycle state (DLM state) is not OEM (LCK\_BOOT, RMA\_RET, RMA\_REQ, RMA\_ACK), AL is determined by DLM state. See [Table 2.6](#).
- When Device Life Cycle state (DLM state) is equal to OEM, AL can be determined by performing authentication through JTAG/SWD. In the MCU, authentication method is the challenge response method with the authentication key installed by user.  
In this case, initial value of AL is determined by PL value (see [Table 2.7](#)).  
See [section 2.13.5.2. Connecting Sequence and JTAG/SWD Authentication](#) for authentication sequence.

In addition, using authentication key AL2 and AL1 can be prohibited by setting Protection LCKS and LCKNS respectively. Setting LCKS and LCKNS can be performed by the boot command. After setting, LCKS can only be cleared by the All erase command, LCKNS can be cleared either by the All erase command or secure access.

Note 1. Both of invasive and non-invasive debug. See [section 2.14. References \[1\]](#) for details about invasive and non-invasive debug.

Description in [Table 2.6](#) and [Table 2.7](#) can also be used for serial Flash programmer.



**Table 2.6 Authentication Level (1)**

Conditions			Available Authentication Level
DLM state	LCKS	LCKNS	
LCK_BOOT	Don't care	Don't care	AL0 (default)
RMA_RET	Don't care	Don't care	AL0 (default)
RMA_REQ	Don't care	Don't care	AL0 (default)
RMA_ACK	Don't care	Don't care	AL2 (default)

**Table 2.7 Authentication Level (2)**

Condition				Available Authentication Level
DLM state	PL	LCKS	LCKNS	
OEM	PL2	Don't care	Don't care	AL2 (default)
OEM	PL1	0b111	Don't care	AL1 (default) AL2 (authentication using AL2_KEY)
OEM	PL1	0b000 (using AL2_KEY is prohibited)	Don't care	AL1 (default)
OEM	PL0	0b111	0b111	AL0 (default) AL1 (authentication using AL1_KEY) AL2 (authentication using AL2_KEY)
OEM	PL0	0b111	0b000 (using AL1_KEY is prohibited)	AL0 (default) AL2 (authentication using AL2_KEY)
OEM	PL0	0b000 (using AL2_KEY is prohibited)	0b111	AL0 (default) AL1 (authentication using AL1_KEY)
OEM	PL0	0b000 (using AL2_KEY is prohibited)	0b000 (using AL1_KEY is prohibited)	AL0 (default)

Note: When debugging using SWJ-DP, set CDBGPWRUPREQ to 1.

Note: Debug connection from external emulator to CoreSight components through Debug APB is enabled as default setting, don't care is the authentication mechanism. However, this connection can be disabled by software programming. See DBGAUTH0.DEVICEEN for details.

### Debug connection while in Boot mode

Debug capability is not available while the MCU is operating in Boot mode.

### 2.7.2 Debug Protection Mechanism by Software

In this mechanism, the challenge and response authentication procedure does not perform, instead software can program to directly enable debug function of the MCU. In other words, debug function is controlled completely by software.

The debug function can be enabled by setting the following bits:

- DBGAUTH0.NIDEN0 = 1: Non-invasive debug enable
- DBGAUTH0.DBGEN0 = 1: Invasive debug enable
- DAUTHCTRL.INTSPNIDEN = 1: Secure non-invasive debug enable. See Reference [1] for details
- DAUTHCTRL.INTSPIDEN = 1: Secure invasive debug enable. See Reference [1] for details.

In addition, external debug access to processor through AHB-AP can also be enabled by writing 1 to DBGAUTH0.DBGENAP and DBGAUTH1.SPIDENAP for Non-secure and Secure access respectively.

See DBGAUTH0 and DBGAUTH1 in [section 2.8.5.3. DBGSTOPCR : Debug Stop Control Register](#) and [section 2.8.5.4. DBGAUTH0 : Debug Authentication Control Register 0](#) for more details.

Note: Authentication through software programming is controlled by OFS1(\_SEC).SWDBG.

Writing 1 to OFS1(\_SEC).SWDBG disables authentication through software programming, writing 0 to OFS1(\_SEC).SWDBG enables authentication through software programming.

Note: Authentication through software programming is not restricted by DLM state, PL, and LCK(N)S.

Note: Except DAUTHCTRL.INTSPNIDEN and DAUTHCTRL.INTSPIDEN overrides processor external Secure invasive/non-invasive debug authentication interface, debug function enabled by authentication through JTAG/SWD cannot be disabled by software.

## 2.7.3 Effect of Debug Function

The debug function affects the inside and outside of CPU. This section describes the effects of debug function excluding debug authentication.

### 2.7.3.1 Low power mode

All CoreSight debug components can store the register settings even when the CPU enters Deep Sleep, Software Standby, or Deep Software Standby mode if SYOCDDBGEN0 is 1 and any of the following is true:

- CDBGPWRUPREQ = 1 and AL is AL2 or AL1
- OFS1(\_SEC).SWDBG is 0 and DBGAUTH0.DBGEN0 or DBGAUTH0.NIDEN0 is 1.

However, AHB-AP cannot respond to On-Chip Debug (OCD) access in these low power modes. The OCD must wait for cancellation of the low power mode to access the CoreSight debug components.

The MCU does not response to On-Chip Debug (OCD) when the OCD starts to connect to the MCU while the MCU is in Software Standby mode or Deep Software Standby mode.

### 2.7.3.2 Resets and interrupts

During On-Chip Debug (OCD) mode, when the CPU enters the break state, IWDT and corresponding WDT are stopped. During OCD mode, the CPU is running the application and debug function AL2 or AL1 is enabled, some resets and interrupts are restrained depending on DBGSTOPPCR setting such as PVDn, WDT, and IWDT. For PVDn (n = 1, 2), interrupt and reset depend on whether DBGSTOPPCR is in the break state or not. Some access errors are not generated by external debugger (DAP) access. For details, see [Table 2.8](#).

**Table 2.8 Controlling of resets and interrupts generation in OCD mode**

Reset and interrupt name	Break state	Running application
RES pin reset	Same as user mode	
Power-on reset	Same as user mode	
Independent watchdog timer reset/interrupt <sup>*2</sup>	Does not occur <sup>*1</sup>	Depends on DBGSTOPPCR setting <sup>*2</sup>
Watchdog timer reset/interrupt	Does not occur <sup>*1</sup>	Depends on DBGSTOPPCR setting <sup>*2</sup>
CPU lockup reset/interrupt	Does not occur	Same as user mode
Voltage monitor 0 reset	Same as user mode	
Voltage monitor 1 reset/interrupt	Depends on DBGSTOPPCR setting <sup>*3</sup>	
Voltage monitor 2 reset/interrupt	Depends on DBGSTOPPCR setting <sup>*3</sup>	
Cache/TCM ECC error BusFaults	Same as user mode	
SRAM parity error reset/interrupt	Same as user mode <sup>*3</sup>	
Deep software standby reset	Same as user mode	
Software reset	Same as user mode	
Bus error reset/interrupt	Same as user mode <sup>*4</sup>	

Note 1. IWDT and WDT always stop in this mode if any of the following is true:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OFS1(\_SEC).SWDBG is 0, and DBGAUTH0.DBGEN0 is 1

Note 2. IWDT and WDT operation depends on DBGSTOPPCR setting if any of the following is true:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OFS1(\_SEC).SWDBG is 0, and DBGAUTH0.DBGEN0 is 1

Note 3. Reset or interrupt masking depends on DBGSTOPPCR setting.

Note 4. The reset request or interrupt request is not generated when an error is detected for DAP access.

## 2.8 Registers Description

### 2.8.1 CPU Control Registers

In addition to Cortex-M85 registers, the MCU also provides additional specific control and status registers that can be considered as CPU control register.

#### 2.8.1.1 CPUSAR : CPU Security Attribution Register

Base address: CPSCU = 0x4000\_8000  
CPSCU\_NS = 0x5000\_8000

Offset address: 0x170

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPUSA0	CPU Control Registers Security Attribution 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

#### CPUSA0 bit (CPU Control Registers Security Attribution)

The CPUSA0 bit is Security attributes for CPU Control registers. The target registers are:

- CPU.CPULCKUPCR
- CPU.CPUCRPT

#### 2.8.1.2 CPULCKUPCR : CPU Lockup Control Register

Base address: CPU\_CTRL = 0x4000\_F000  
CPU\_CTRL\_NS = 0x5000\_F000

Offset address: 0x030

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after detection of CPU lockup 0: Non-maskable Interrupt 1: CPU Lockup reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (CPUSAR.CPUSA0), P-TYPE-2

When CPU enters the lockup state, it can be selected to generate either a CPU Lockup reset or an interrupt<sup>\*1</sup> by writing to CPULCKUPCR.OAD.

The CPULCKUPCR security attribution is controlled by CPUSAR. The AIRCR.BFHFNMINs and CPUSAR.CPUSA0 bits are recommended to be set to the same security attribution.

Writing to CPULCKUPCR.OAD bit is protected by CPU Control Register Protect Register (CPUCRPT).

Note 1. If lockup state occurs from a Secure HardFault when AIRCR.BFHFNMINs is set to 1 or the NMI handler, an NMI does not cause the processor exit lockup state. See [section 2.14. References \[4\]](#).

### 2.8.1.3 CPUCRPT : CPU Control Register Protection Register

Base address: CPU\_CTRL = 0x4000\_F000 (Secure)  
CPU\_CTRL\_NS = 0x5000\_F000 (Non-secure)

Offset address: 0x840

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]														PROTECT	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Writing to CPULCKUPCR register is permitted. 1: Writing to CPULCKUPCR register is not permitted. Read access is permitted.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 in KEY[7:0] bits at the same time. When values other than 0xA5 are written to KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.	W

Note: S-TYPE-3 (CPUSAR.CPUSA0), P-TYPE-2

The CPUCRPT.PROTECT bit enables writing to CPULCKUPCR.

The security attribution of CPUCRPT is controlled by CPUSAR.

### 2.8.1.4 CPULOCKCR : CPU Function Lock Control Register

Base address: CPU\_CTRL = 0x4000\_F000 (Secure)

Offset address: 0x400

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	LCKD CAIC	LCKD TGU	LCKIT GU	LCKS AU	LCKS MPU	LCKS VTAIR
Value after reset :	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LCKSVTAIR	Disable writes to the following secure registers from software or from a debug agent that is connected to the processor: <b>VTOR_S, AIRCR.PRIS, AIRCR.BFHFNMINs</b>	R/W <sup>1</sup>
1	LCKSMPU	Disable writes to registers that are associated with the Secure MPU region from software or from a debug agent connected to the processor: <b>MPU_CTRL, MPU_RNR, MPU_RBAR, MPU_RLAR, MPU_RBAR_An, MPU_RLAR_An</b>	R/W <sup>1</sup>
2	LCKSAU	Disable writes to registers that are associated with the SAU region from software or from a debug agent connected to the processor: <b>SAU_CTRL, SAU_RNR, SAU_RBAR, SAU_RLAR</b>	R/W <sup>1</sup>
3	LCKITGU	Disable writes to registers that are associated with the ITCM interface security gating from software or from a debug agent connected to the processor: <b>ITGUCTRL, ITGU_LUTn</b>	R/W <sup>1</sup>

Bit	Symbol	Function	R/W
4	LCKDTGU	Disable writes to registers that are associated with the DTCM interface security gating from software or from a debug agent connected to the processor: <b>DTGUCTRL, DTGU_LUTn</b>	R/W <sup>1</sup>
5	LCKDCAIC	Disable access to the instruction cache direct cache access registers <b>DCAICLR</b> and <b>DCAICRR</b>	R/W <sup>1</sup>
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2

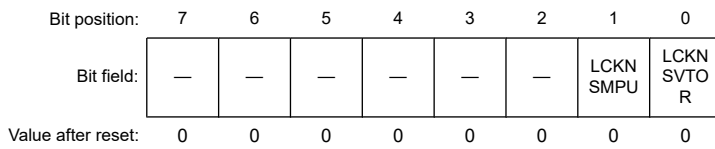
Note 1. Writing 0 is ignored.

This register controls some miscellaneous processor signals. Reset is the only clear condition.

### 2.8.1.5 CPULOCKCRNS : CPU Non-secure Function Lock Control Register

Base address CPU\_CTRL\_NS = 0x5000\_F000 (Non-secure)

Offset address: 0x500



Bit	Symbol	Function	R/W
0	LCKNSVTOR	Disable writes to the <b>VTOR_NS</b> register	R/W <sup>1</sup>
1	LCKNSMPU	Disable writes to registers that are associated with the Non-secure MPU region from software or from a debug agent connected to the processor: <b>MPU_CTRL_NS, MPU_RNR_NS, MPU_RBAR_NS, MPU_RLAR_NS, MPU_RBAR_A_NSn, MPU_RLAR_A_NSn</b>	R/W <sup>1</sup>
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-7, P-TYPE-2

Note 1. Writing 0 is ignored.

This register controls some miscellaneous processor signals. Reset is the only clear condition.

### 2.8.2 Address Spaces for External Debugger

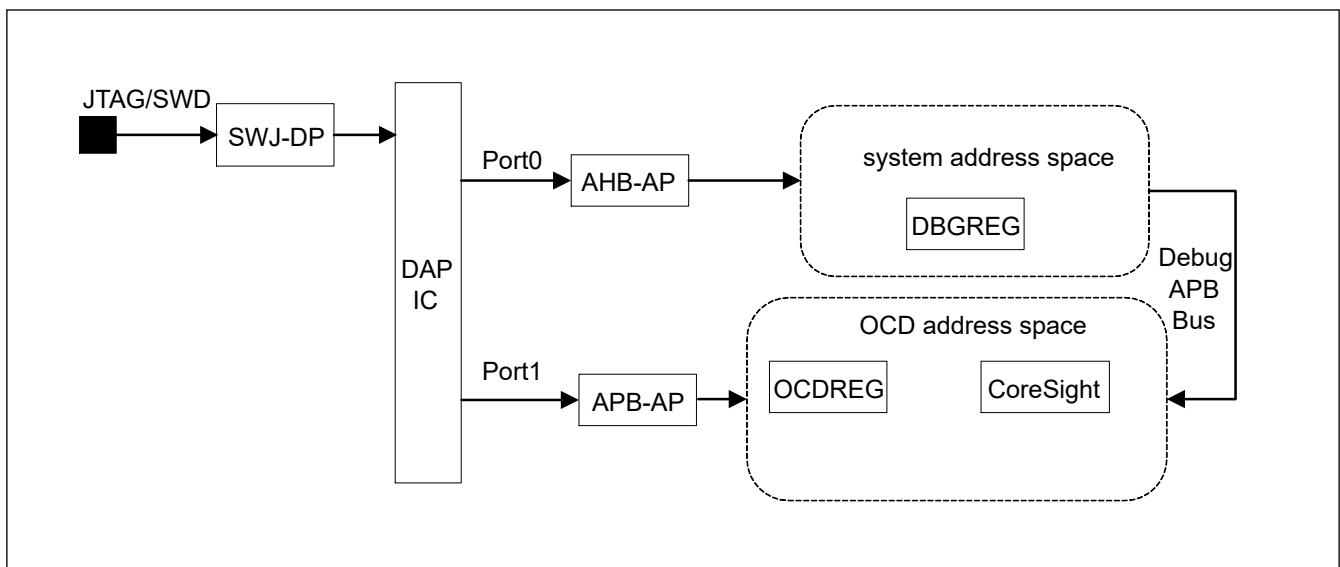


Figure 2.2 Block diagram of the AP connection and address spaces

The MCU debug system includes two CoreSight Access Ports (AP):

- AHB-AP, which is connected to the CPU bus matrix and has the same access to the system address space as the CPU
- APB-AP, which has OCD address space and is connected to the Coresight components and OCDREG registers.

Figure 2.2 shows a block diagram of the AP connection and address spaces. For debug purpose, there are two address spaces, DBGREG and OCDREG.

DBGREG is located in the system address space and can be accessed from an OCD emulator, a CPU, or other bus masters in the MCU.

OCDREG is located in the OCD address space and can be accessed from an OCD emulator, a CPU, or other bus masters in the MCU. See Table 2.10 for details.

### 2.8.3 CPU Peripheral

CoreSight components are accessible from either CPU or OCD emulator. Table 2.9 shows dedicated Peripheral address of CPU. Table 2.10, Table 2.11, and Table 2.12 describe CoreSight component address.

**Table 2.9 CPU peripherals**

Component name	Start address	End address	Note
ITM	0xE000_0000	0xE000_0FFF	See section 2.14. References [2]
DWT	0xE000_1000	0xE000_1FFF	See section 2.14. References [2]
BPU	0xE000_2000	0xE000_2FFF	See section 2.14. References [2]
PMU	0xE000_3000	0xE000_3FFF	See section 2.14. References [2]
SCS	0xE000_E000	0xE000_EFFF	See section 2.14. References [2]
SCS Non-secure alias	0xE002_E000	0xE002_EFFF	See section 2.14. References [2]
TPIU <sup>*1</sup>	0xE004_0000	0xE004_0FFF	See section 2.14. References [3]
ETM	0xE004_1000	0xE004_1FFF	See section 2.14. References [2]
CTIO	0xE004_2000	0xE004_2FFF	See section 2.14. References [5]
EPPB ROM Table	0xE00F_E000	0xE00F_EFFF	See section 2.14. References [3]
Processor ROM Table	0xE00F_F000	0xE00F_FFFF	See section 2.14. References [3]

Note 1. TPIU registers should not be accessed without connection of OCD emulator to avoid stopping bus access.

**Table 2.10 CPU peripherals (Secure CPU access view)**

Component name	Start address	End address	Note
System ROM Table	0x4001_0000	0x4001_0FFF	See section 2.14. References [3]
OCDREG	0x4001_1000	0x4001_1FFF	See section 2.14. References [5]
CTI	0x4001_2000	0x4001_2FFF	See section 2.14. References [5]
Funnel	0x4001_3000	0x4001_3FFF	See section 2.14. References [5]
TMC(ETB)	0x4001_4000	0x4001_4FFF	See section 2.14. References [5]
Time Stamp Generator	0x4001_5000	0x4001_5FFF	See section 2.14. References [5]

Note: These registers can only be accessed from CPU when DbgSwEnable of APB Control/Status Word register is 1. See section 2.14. References [5].

**Table 2.11 CPU peripherals (Non-secure CPU access view)**

Component name	Start address	End address	Note
System ROM Table	0x5001_0000	0x5001_0FFF	See section 2.14. References [3]
OCDREG	0x5001_1000	0x5001_1FFF	See section 2.14. References [5]
CTI	0x5001_2000	0x5001_2FFF	See section 2.14. References [5]
Funnel	0x5001_3000	0x5001_3FFF	See section 2.14. References [5]
TMC(ETB)	0x5001_4000	0x5001_4FFF	See section 2.14. References [5]
Time Stamp Generator	0x5001_5000	0x5001_5FFF	See section 2.14. References [5]

Note: These registers can only be accessed from CPU when DbgSwEnable of APB Control/Status Word register is 1. See [section 2.14. References \[5\]](#).

**Table 2.12 CPU peripherals (OCD emulator access view)**

Component name	Start address	End address	Note
System ROM Table	0x8001_0000	0x8001_0FFF	See <a href="#">section 2.14. References [3]</a>
OCDREG	0x8001_1000	0x8001_1FFF	See <a href="#">section 2.14. References [5]</a>
CTI	0x8001_2000	0x8001_2FFF	See <a href="#">section 2.14. References [5]</a>
Funnel	0x8001_3000	0x8001_3FFF	See <a href="#">section 2.14. References [5]</a>
TMC(ETB)	0x8001_4000	0x8001_4FFF	See <a href="#">section 2.14. References [5]</a>
Time Stamp Generator	0x8001_5000	0x8001_5FFF	See <a href="#">section 2.14. References [5]</a>

## 2.8.4 ROM Table

The MCU has three ROM Tables:

- A processor ROM Table
- An EPPB ROM Table for the Cortex-M85 CPU
- A system ROM Table

The EPPB ROM Table points to the TPIU component and processor ROM Table. The EPPB ROM Table does not point to or be pointed by system ROM Table.

The processor ROM Table contains entries that hold a list of debug components that are implemented in Cortex-M85. The system ROM Table contains entries that hold a list of debug components implemented as CoreSight debug components of the MCU.

### 2.8.4.1 ROM entries

ROM entries holds a list of CoreSight debug components in the system. The OCD emulator can use the ROM entries to determine which components are implemented in the system.

The entries of Debug components are described as bellow.

[Table 2.13: System ROM entries](#)

[Table 2.14: EPPB ROM entries](#)

[Table 2.15: Processor ROM entries \(see \[section 2.14. References\]\(#\)\).](#)

**Table 2.13 System ROM entries**

#	Address <sup>*1</sup>	Access size	R/W	Value	Target module
0	0x4001_0000	32 bits	Read-only	0x00001003	OCDREG
1	0x4001_0004	32 bits	Read-only	0x00002003	CTI
2	0x4001_0008	32 bits	Read-only	0x00003003	Funnel
3	0x4001_000C	32 bits	Read-only	0x00004003	ETB
4	0x4001_0010	32 bits	Read-only	0x00005003	TSG
5	0x4001_0014	32 bits	Read-only	0x00000000	(End of entries)

Note 1. The base address 0x4001\_0000 is dedicated for CPU secure access.  
 The base address for CPU non-secure access is 0x5001\_0000.  
 The base address for OCD emulator access is 0x8001\_0000.

**Table 2.14 EPPB ROM entries (1 of 2)**

#	Address	Access size	R/W	Value	Target module
0	0xE00F_E000	32 bits	Read-only	0x00001003	Processor ROM Table
1	0xE00F_E004	32 bits	Read-only	0xFFFF42003	TPIU

**Table 2.14 EPPB ROM entries (2 of 2)**

#	Address	Access size	R/W	Value	Target module
2	0xE00F_E008	32 bits	Read-only	0x00000000	(End of entries)

**Table 2.15 Processor ROM entries**

#	Address	Access size	R/W	Value	Target module
0	0xE00F_F000	32 bits	Read-only	0xFFF0F003	SCS
1	0xE00F_F004	32 bits	Read-only	0xFFF02003	DWT
2	0xE00F_F008	32 bits	Read-only	0xFFF03003	BPU
3	0xE00F_F00C	32 bits	Read-only	0xFFF01003	ITM
5	0xE00F_F014	32 bits	Read-only	0xFFF42003	ETM
6	0xE00F_F018	32 bits	Read-only	0xFFF04003	PMU
7	0xE00F_F01C	32 bits	Read-only	0xFFF43003	CTIO
8	0xE00F_F020	32 bits	Read-only	0xFFF47002	PMC
9	0xE00F_F024	32 bits	Read-only	0x00000000	(End of entries)

### 2.8.4.2 CoreSight Component Registers

ROM Table has CoreSight Component Register defined in Arm CoreSight Architecture.

[Table 2.16](#) shows the registers of system ROM Table.

[Table 2.17](#) shows processor EPPB ROM Table.

[Table 2.18](#) shows processor ROM Table. See [section 2.14. References \[2\]](#).

**Table 2.16 Registers of system ROM Table**

Name	Address	Access size	R/W	Initial value
PID4	0x8001_0FD0	32 bits	Read-only	0x00000004
PID5	0x8001_0FD4	32 bits	Read-only	0x00000000
PID6	0x8001_0FD8	32 bits	Read-only	0x00000000
PID7	0x8001_0FDC	32 bits	Read-only	0x00000000
PID0	0x8001_0FE0	32 bits	Read-only	0x0000003B
PID1	0x8001_0FE4	32 bits	Read-only	0x00000030
PID2	0x8001_0FE8	32 bits	Read-only	0x0000000A
PID3	0x8001_0FEC	32 bits	Read-only	0x00000000
CID0	0x8001_0FF0	32 bits	Read-only	0x0000000D
CID1	0x8001_0FF4	32 bits	Read-only	0x00000010
CID2	0x8001_0FF8	32 bits	Read-only	0x00000005
CID3	0x8001_0FFC	32 bits	Read-only	0x000000B1

**Table 2.17 Processor EPPB ROM Table (1 of 2)**

Name	Address	Access size	R/W	Initial value
PID4	0xE00F_EFD0	32 bits	Read-only	0x00000004
PID5	0xE00F_EFD4	32 bits	Read-only	0x00000000
PID6	0xE00F_EFD8	32 bits	Read-only	0x00000000
PID7	0xE00F_EFDC	32 bits	Read-only	0x00000000
PID0	0xE00F_EFE0	32 bits	Read-only	0x0000003B
PID1	0xE00F_EFE4	32 bits	Read-only	0x00000030



**Table 2.17 Processor EPPB ROM Table (2 of 2)**

Name	Address	Access size	R/W	Initial value
PID2	0xE00F_EFE8	32 bits	Read-only	0x0000000A
PID3	0xE00F_EFEC	32 bits	Read-only	0x00000000
CID0	0xE00F_EFF0	32 bits	Read-only	0x0000000D
CID1	0xE00F_EFF4	32 bits	Read-only	0x00000010
CID2	0xE00F_EFF8	32 bits	Read-only	0x00000005
CID3	0xE00F_EFFC	32 bits	Read-only	0x000000B1

**Table 2.18 Processor ROM Table**

Name	Address	Access size	R/W	Initial value
PID4	0xE00F_FFD0	32 bits	Read-only	0x00000004
PID5	0xE00F_FFD4	32 bits	Read-only	0x00000000
PID6	0xE00F_FFD8	32 bits	Read-only	0x00000000
PID7	0xE00F_FFDC	32 bits	Read-only	0x00000000
PID0	0xE00F_FFE0	32 bits	Read-only	0x000000D4
PID1	0xE00F_FFE4	32 bits	Read-only	0x000000B4
PID2	0xE00F_FFE8	32 bits	Read-only	0x0000000B
PID3	0xE00F_FFEC	32 bits	Read-only	0x00000000
CID0	0xE00F_FFF0	32 bits	Read-only	0x0000000D
CID1	0xE00F_FFF4	32 bits	Read-only	0x00000010
CID2	0xE00F_FFF8	32 bits	Read-only	0x00000005
CID3	0xE00F_FFFC	32 bits	Read-only	0x000000B1

### 2.8.5 DBGREG

Base\_Address: CPU\_DBG = 0x4001\_B000

The DBGREG module controls the debug functionality.

[Table 2.19](#) shows a list of DBGREG registers.

**Table 2.19 DBGREG registers**

Name	Symbol	DAP port	Address	Access size	R/W
Debug Status Register	DBGSTR	Port 0	0x4001_B000	32 bits	Read-only
Debug Stop Control Register	DBGSTOPCR	Port 0	0x4001_B010	32 bits	Read/Write
Debug Authentication Control Register0	DBGAUTH0	Port 0	0x4001_B020	32 bits	Read/Write
Debug Authentication Control Register1	DBGAUTH1	Port 0	0x4001_B024	32 bits	Read/Write
Trace Port Control Register	TRPORTCR	Port 0	0x4001_B030	32 bits	Read/Write
Cache Debug Control Register	CACHEDBGCR	Port 0	0x4001_B040	32 bits	Read/Write
MOCO Enable Register for Debug	DBGMOCOEN	Port 0	0x4001_B300	32 bits	Read/Write
Flash Sequencer Clock Select Register for Debug	DBGCLKSEL	Port 0	0x4001_B310	32 bits	Read/Write

### 2.8.5.1 DEBUGSAR : Debug Security Attribution Register

Base address: CPSCU = 0x4000\_8000  
 CPSCU\_NS = 0x5000\_8000

Offset address: 0x180

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DBGS A0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DBGSA0	Debug Resources Security Attribution 0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

#### DBGSA0 bit (Debug Resources Security Attribution 0)

Security attributes of registers for Debug Control. The target registers are as follow:

- DBGREG.DBGAUTH0
- The CoreSight registers connected to APB-AP and mapped to the system address space

Note: This does not affect access from external to APB-AP.

### 2.8.5.2 DBGSTR : Debug Status Register

Base address: CPU\_DBG = 0x4001\_B000  
 CPU\_DBG\_NS = 0x5001\_B000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	CDBG PWRU PACK	CDBG PWRU PREQ	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
27:0	—	These bits are read as 0.	R
28	CDBGPWRUPREQ	Debug power-up request 0: OCD is not requesting debug power-up 1: OCD is requesting debug power-up	R
29	CDBGPWRUPACK	Debug power-up acknowledge 0: Debug power-up request is not acknowledged 1: Debug power-up request is acknowledged	R
31:30	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2

The DBGSTR register is a debug status register to notify OCD status to programs. This register is monitoring Debug connecting state.

### 2.8.5.3 DBGSTOPCR : Debug Stop Control Register

Base address: CPU\_DBG = 0x4001\_B000  
CPU\_DBG\_NS = 0x5001\_B000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	DBGS TOP_ RER	—	—	—	—	—	—	DBGS TOP_ PVD	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DBGS TOP_ WDT0	DBGS TOP_ I WDT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	DBGSTOP_IWDT	Mask bit for IWDT reset/interrupt in the OCD run mode In the OCD break mode, the reset/interrupt is masked and IWDT counter is stopped, regardless of this bit value. 0: Enable IWDT reset/interrupt 1: Mask IWDT reset/interrupt and stop IWDT counter	R/W
1	DBGSTOP_WDT0	Mask bit for WDT reset/interrupt in the OCD run mode In the OCD break mode, the reset/interrupt is masked and WDT counter is stopped, regardless of this bit value. 0: Enable WDT reset/interrupt 1: Mask WDT reset/interrupt and stop WDT counter	R/W
16:2	—	These bits are read as 0. The write value should be 0.	R/W
17	DBGSTOP_PVD	Mask bit for PVDn (n = 1, 2) reset/interrupt 0: Enable PVDn (n = 1, 2) reset/interrupt 1: Mask PVDn (n = 1, 2) reset/interrupt	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	DBGSTOP_RER	Mask bit for SRAM parity error reset/interrupt 0: Enable SRAM parity error reset/interrupt 1: Mask SRAM parity error reset/interrupt	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2

Debug Stop Control Register (DBGSTOPCR) specifies functional stop during debugging.

This register is valid only when any of the following conditions are true:

- CDBGPWRUPREQ is 1, and AL is AL2 or AL1
- OFS1(\_SEC).SWDBG is 0, and DBGAUTH0.DBGEN0 is 1

When none of the above conditions are true, the MCU ignores this register setting and considers all bits to be 0.

## 2.8.5.4 DBGAUTH0 : Debug Authentication Control Register 0

Base address: CPU\_DBG = 0x4001\_B000  
CPU\_DBG\_NS = 0x5001\_B000

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SWDBG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DEVICEEN
Value after reset:	OFS1(_SEC).SWDBG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DBGENAP	—	—	—	NIDEN0	—	—	—	DBGEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DBGEN0*1	CPU invasive debug enable 0: Disabled 1: Enabled	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	NIDEN0	CPU non-invasive debug enable 0: Disabled 1: Enabled	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	DBGENAP*1	CPU AHB-AP (AP0) debug enable 0: Disabled 1: Enabled	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	DEVICEEN*2	APB-AP (AP1) authentication 0: Disabled 1: Enabled	R/W
30:17	—	These bits are read as 0. The write value should be 0.	R/W
31	SWDBG	Software control of debug function 0: Disabled 1: Enabled	R/W

Note: S-TYPE-3 (DEBUGSAR.DBGSA0), P-TYPE-2

Note 1. When AL2 or AL1, the MCU ignores this bit and considers as if it is set to 1.

Note 2. When AL2, the MCU ignores this bit and considers as if it is set to 1.

When software control of debug function (OFS1(\_SEC).SWDBG=0) is enabled, non-secure invasive and non-invasive debug function of processor enabling can be controlled by DBGEN0 and NIDEN0 bits of the register.

Non-secure AHB-AP access can be enabled by DBGENAP bit of the register.

Access from APB-AP to Coresight components through Debug APB can be enabled by DEVICEEN bit of the register.

When OFS1(\_SEC).SWDBG is 1, the MCU ignores this register and considers as if it is set to its initial value.

Note: MCU behaves according to DBGSTOPCR, SYOCDRCR, MCUCTRL when DBGEN0 = 1, and also according to TRPORTCR when (DBGEN0 = 1 or NIDEN0 = 1) is true.

### 2.8.5.5 DBGAUTH1 : Debug Authentication Control Register 1

Base address: CPU\_DBG = 0x4001\_B000

Offset address: 0x024

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

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	SPIDENAP*1	CPU AHB-AP (AP0) debug enable 0: Disabled 1: Enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2

Note 1. When AL2, the MCU ignores this bit and considers as if it is set to 1.

When software control of debug function (OFS1(\_SEC).SWDBG = 0) is enabled, secure access from AHB-AP port to D-AHB can be enabled by SPIDENAP bit of the register.

When OFS1(\_SEC).SWDBG is 1, the MCU ignores this register and considers as if it is set to its initial value.

### 2.8.5.6 CACHEDBGCR : Cache Debug Control Register

Base address: CPU\_DBG = 0x4001\_B000

Offset address: 0x040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	L1RSTDIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	L1RSTDIS	Disable L1 cache automatic invalidation 0: Enable automatic invalidation of the L1 cache 1: Disable automatic invalidation of the L1 cache	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2

The CACHEDBGCR register can control Disable L1 cache invalidation out of reset during debugging.

Note: The setting of the register is valid only if any of the following conditions apply:

- CDBGPWRUPREQ = 1 and AL is AL2 or AL1
- OFS1(\_SEC).SWDBG is 0 and DBGAUTH0.DBGEN0 or DBGAUTH0.NIDEN0 is 1

Note: See section 4.6 Cache Initialization Signal in [section 2.14. References \[2\]](#) before writing to this bit.

### 2.8.5.7 TRPORTCR : Trace Port Control Register

Base address: CPU\_DBG = 0x4001\_B000  
CPU\_DBG\_NS = 0x5001\_B000

Offset address: 0x030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DRV[1:0]	—	—	OE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
0	OE	Data Out Enable bit indicates whether Trace Clock, Trace Data, and SWO outputs are enabled. 0: Output disabled 1: Output enabled	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
3:2	DRV[1:0]	Port Drive Capability Control indicates trace port buffer speed: 0 0: Low-drive 0 1: Middle-drive 1 0: High-speed high-drive 1 1: High-drive	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2

The TRPORTCR register controls trace port of tracing data.

Note: The setting of the register is valid only if any of the following conditions apply:

- CDBGPWRUPREQ = 1, and AL is AL2 or AL1.
- OFS1(\_SEC).SWDBG is 0, and DBGAUTH0.DBGEN0 or DBGAUTH0.NIDEN0 is 1.

### 2.8.5.8 DBGMOCOEN : MOCO Enable Request Register for Debug

Base address: CPU\_DBG = 0x4001\_B000  
CPU\_DBG\_NS = 0x5001\_B000

Offset address: 0x300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MOCO EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MOCOEN	MOCO enable request 0: No request MOCO enable 1: Request MOCO enable	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2

The DBGMOCOEN register enables MOCO operation. When external debugger is not connected, the register does not have any effect. During P/E mode, the value of this register cannot be updated.

**Note:** Usage of the register:

When only non-secure debug function is enabled, the debugger cannot set clock-related registers that are protected by TrustZone. Therefore, writing to flash memory (by setting software breakpoint) is not available if the FCLK frequency is not at minimum required speed for programming flash operation.

In this case, by setting DBGMOCOEN and DBGFCLKSEL, the flash sequency clock can be changed to MOCO to enable writes to flash memory.

### 2.8.5.9 DBGFCLKSEL : Flash Sequencer Clock Select Register for Debug

Base address: CPU\_DBG = 0x4001\_B000  
CPU\_DBG\_NS = 0x5001\_B000

Offset address: 0x310

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FCLK SEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FCLKSEL	Flash sequencer clock select 0: FCLK 1: MOCO	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

**Note:** S-TYPE-5, P-TYPE-2

The DBGFCLKSEL register switches the flash P/E sequencer clock. When external debugger is not connected, the register does not have any effect. During P/E mode, the value of this register cannot be updated.

Flash P/E is available with MOCO instead of FCLK even when FCLK is not in the range to support flash P/E. The following flow shows how to use MOCO instead of FCLK.

1. Wait until CPU is in debug state (DHCSR.S\_HALT = 1).
2. Check that the flash is not in P/E mode (when flash is in P/E mode, flash P/E by the external host cannot be performed.)
3. Set DBGMOCOEN.MOCOEN to 1.
4. Wait for MOCO oscillation settling time.
5. Set DBGFCLKSEL.FCLKSEL to 1.
6. Perform flash P/E.
7. Clear DBGFCLKSEL.FCLKSEL to 0.
8. Clear DBGMOCOEN.MOCOEN to 0.
9. Restart the CPU.

**Note:** Usage of the register:

When only non-secure debug function is enabled, the debugger cannot set clock-related registers that are protected by TrustZone. Therefore, writing to flash memory (by setting software breakpoint) is not available if the FCLK frequency is not at minimum required speed for programming flash operation.

In this case, by setting DBGMOCOEN and DBGFCLKSEL, the flash sequency clock can be changed to MOCO to enable writes to flash memory.

### 2.8.5.10 CoreSight Component Registers

DBGREG has CoreSight Component Registers defined in ARM CoreSight Architecture. [Table 2.20](#) shows the list of the registers.

**Table 2.20 CoreSight component registers**

Name	Address	Access size	R/W	Initial value
PID4	0x4001_BFD0	32 bits	Read-only	0x00000004
PID5	0x4001_BFD4	32 bits	Read-only	0x00000000
PID6	0x4001_BFD8	32 bits	Read-only	0x00000000
PID7	0x4001_BFDC	32 bits	Read-only	0x00000000
PID0	0x4001_BFE0	32 bits	Read-only	0x00000005
PID1	0x4001_BFE4	32 bits	Read-only	0x00000030
PID2	0x4001_BFE8	32 bits	Read-only	0x0000000A
PID3	0x4001_BFEC	32 bits	Read-only	0x00000000
CID0	0x4001_BFF0	32 bits	Read-only	0x0000000D
CID1	0x4001_BFF4	32 bits	Read-only	0x000000F0
CID2	0x4001_BFF8	32 bits	Read-only	0x00000005
CID3	0x4001_BFFC	32 bits	Read-only	0x000000B1

Note: S-TYPE-5, P-TYPE-2

### 2.8.6 OCDREG

OCDREG is a register module provided for OCD (On-Chip Debug) Emulator functionalities. OCDREG is implemented as a CoreSight compliant component. [Table 2.21](#) shows register list of OCDREG.

**Table 2.21 Register list of OCDREG**

Name	Symbol	DAP port	OCD Address	CPU Address	Access size	R/W
MCU Status Register	MCUSTAT	Port 1	0x8000_0400	Cannot be accessed from CPU	32 bits	Read only
MCU Control Register	MCUCTRL	Port 1	0x8001_1004	0x4001_1004 (Secure) 0x5001_1004 (Non-secure)	32 bits	Read/Write
JTAG Boot Mode Entry Register	JBMDR	Port 1	0x8001_1100	0x4001_1100 (Secure) 0x5001_1100 (Non-secure)	32 bits	Read/Write
JTAG Boot Receive Data Register	JBRDR	Port 1	0x8001_1120	0x4001_1120 (Secure) 0x5001_1120 (Non-secure)	32 bits	Read/Write
JTAG Boot Transmit Data Register	JBTDNR	Port 1	0x8001_1130	0x4001_1130 (Secure) 0x5001_1130 (Non-secure)	32 bits	Read/Write
JTAG Boot Status Register	JBSTR	Port 1	0x8001_1140	0x4001_1140 (Secure) 0x5001_1140 (Non-secure)	32 bits	Read/Write
JTAG Boot Interrupt Control Register	JBICR	Port 1	0x8001_1150	0x4001_1150 (Secure) 0x5001_1150 (Non-secure)	32 bits	Read/Write

Note: Accesses to OCDREG from CPU is only permitted by privileged access.  
Accesses to OCDREG from external debug through APB-AP is permitted by either Privileged access or Unprivileged access.



Note: Do not write to MCUCTRL using JTAG/SWD while the CPU is accessing MCUCTRL, JBMDR, JBRDR, JBTDR or JBICR.  
 Do not write to JBMDR using JTAG/SWD except during RES pin reset.  
 Do not write to JBRDR using JTAG/SWD while the CPU is accessing MCUCTRL, JBMDR, JBRDR, JBTDR or JBICR.  
 Do not write to JBTDR using JTAG/SWD.  
 Do not write to JBSTR using JTAG/SWD.  
 Do not write to JBICR using JTAG/SWD.

### 2.8.6.1 MCUSTAT : MCU Status Register

Base address: 0x8000\_0400\*1

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	DSTB Y3	DSTB Y2	DSTB Y1	—	—	—	STBY1
Value after reset:	0	0	0	0	0	0	0	0	0	x	x	x	0	0	x	x
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	AL[1:0]	—	—	—	—	—	—	—	—	—	—	CPUS TOPCLK	SLEEP	—
Value after reset:	1	0	0	0	0	0	0	1	0	0	0	0	0	x	x	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R
1	SLEEP	Indicate CPU is in CPU Sleep mode 0: CPU is not in CPU Sleep mode 1: CPU is in CPU Sleep mode	R
2	CPUSTOPCLK	Indicate CPU clock is stopped 0: CPU clock is not stopped. It indicates that the MCU is in Normal or CPU Sleep mode. 1: CPU clock is stopped. It indicates that the MCU is in CPU Deep Sleep mode, Software Standby mode, or Deep Software Standby mode.	R
7:3	—	These bits are read as 0.	R
8	—	This bit is read as 1.	R
11:9	—	These bits are read as 0.	R
13:12	AL[1:0]	AL monitor 0x0: AL0 0x1: AL1 0x3: AL2	R
14	—	This bit is read as 0.	R
15	—	This bit is read as 1.	R
16	STBY1	Indicate the MCU is in Software Standby mode 0: MCU is not in Software Standby mode 1: MCU is in Software Standby mode	R
19:17	—	These bits are read as 0.	R
20	DSTBY1	Indicate the MCU is in Deep Software Standby mode 1 0: MCU is not in Deep Software Standby mode 1 1: MCU is in Deep Software Standby mode 1	R
21	DSTBY2	Indicate the MCU is in Deep Software Standby mode 2 0: MCU is not in Deep Software Standby mode 2 1: MCU is in Deep Software Standby mode 2	R
22	DSTBY3	Indicate the MCU is in Deep Software Standby mode 3 0: MCU is not in Deep Software Standby mode 3 1: MCU is in Deep Software Standby mode 3	R
31:23	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2

Note 1. The address of MCUSTAT register can only be accessed by external debugger.

The MCUSTAT register indicates the MCU status and authentication status. All bits in the MCUSTAT register are monitoring bits, so they cannot be reset. The register has no security protection.

### 2.8.6.2 MCUCTRL : MCU Control Register

Base address: OCD\_CPU = 0x4001\_1000  
 OCD\_CPU\_NS = 0x5001\_1000  
 OCD\_DAP = 0x8001\_1000<sup>\*1</sup>

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPUWAIT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EDBGRQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EDBGRQ	External Debug Request. Writing 1 to the bit causes a CPU Halt or Debug Monitor exception request. 0: Debug event is not requested 1: Debug event is requested	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
16	CPUWAIT	CPU Wait Setting 0: Deassert CPUWAIT CPU starts boot-up sequence and instruction execution out of Reset. 1: Assert CPUWAIT CPU is forced into a quiescent state out of Reset.	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2  
 Note: This register is valid only when any of the following conditions are true:

- CDBGPWUPREQ is 1, and AL is AL2 or AL1.
- OFS1(\_SEC).SWDBG is 0, and DBGAUTH0.DBGEN0 is 1.

When none of the above conditions are true, the MCU ignores this register setting and considers all bits to be 0.

Note 1. This address is accessed from external debugger.

The MCUCTRL register provides the external debugger ability to enter debug state by external event.

### 2.8.6.3 Authentication Control Registers

Authentication registers accessible from debugger are placed in OCDREG.

#### 2.8.6.3.1 JBMDR : JTAG Boot Mode Entry Register

Base address: OCD\_CPU = 0x4001\_1000  
 OCD\_CPU\_NS = 0x5001\_1000  
 OCD\_DAP = 0x8001\_1000<sup>\*1</sup>

Offset address: 0x100

Bit position:	31															7	0																
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	KEY[7:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	KEY[7:0]	Mode entry key Pin reset releases after 0xA5 is set, then system enters the JTAG boot mode. MDSR.JBOTS = 1 when system transitions to JTAG boot mode.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2

Note 1. This address is accessed from external debugger.

The JBMDR register sets command from the debugger.

This register is initialized by the factors listed Table 5.2 in section 5, Resets, but also initialized by debugger disconnection.

### 2.8.6.3.2 JBRDR : JTAG Boot Receive Data Register

Base address: OCD\_CPU = 0x4001\_1000  
OCD\_CPU\_NS = 0x5001\_1000  
OCD\_DAP = 0x8001\_1000\*1

Offset address: 0x120

Bit position: 31

0

Bit field:

RDAT[31:0]

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	RDAT[31:0]	Received data register R/W is possible for both the external host and CPU, but the following usage is recommended: W: External host R: CPU (Boot firmware) When JBSTR.RDF = 1, write is not possible and an error occurs.	R/W

Note: S-TYPE-5, P-TYPE-2

Note 1. This address is accessed from external debugger.

The JBRDR register is used for receiving data from the debugger.

### 2.8.6.3.3 JBTDR : JTAG Boot Transmit Data Register

Base address: OCD\_CPU = 0x4001\_1000  
OCD\_CPU\_NS = 0x5001\_1000  
OCD\_DAP = 0x8001\_1000\*1

Offset address: 0x130

Bit position: 31

0

Bit field:

TDAT[31:0]

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	TDAT[31:0]	Transmitted data register R/W is possible for both the external host and CPU, but the following usage is recommended: W: CPU (Boot firmware) R: External host When JBSTR.TDE = 1, read is not possible and an error occurs.	R/W

Note: S-TYPE-5, P-TYPE-2

Note 1. This address is accessed from external debugger.

The JBTDR register is used for transmitting data to the debugger.

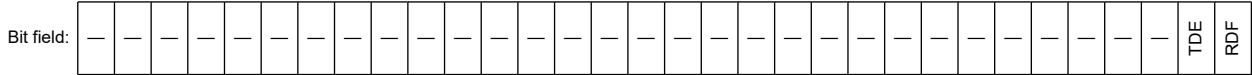
### 2.8.6.3.4 JBSTR : JTAG Boot Status Register

Base address: OCD\_CPU = 0x4001\_1000  
 OCD\_CPU\_NS = 0x5001\_1000  
 OCD\_DAP = 0x8001\_1000\*1

Offset address: 0x140

Bit position: 31

1 0



Value after reset: 0 1 0

Bit	Symbol	Function	R/W
0	RDF	Receive buffer full 0: No receiving data 1: There is receiving data	R/W
1	TDE	Transmit data empty 0: There is data transmission 1: No data transmission	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2

Note 1. This address is accessed from external debugger.

The JBSTR register is used for monitoring booting status.

#### RDF flag (Receive buffer full)

The RDF bit indicates receive buffer is full or not.

[Set conditions]

- Read access to JBTDR
- Write 1 to JBSTR.RDF

[Clear conditions]

- Write access to JBTDR
- Write 0 to JBSTR.RDF

#### TDE flag (Transmit data empty)

The TDE bit indicates transmit data is empty or not.

[Set conditions]

- Read access to JBTDR
- Write 1 to JBSTR.TDE

[Clear conditions]

- Write access to JBTDR
- Write 0 to JBSTR.TDE

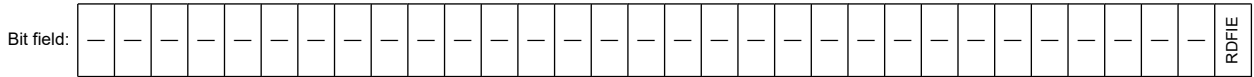
### 2.8.6.3.5 JBICR : JTAG Boot Interrupt Control Register

Base address: OCD\_CPU = 0x4001\_1000  
 OCD\_CPU\_NS = 0x5001\_1000  
 OCD\_DAP = 0x8001\_1000\*1

Offset address: 0x150

Bit position: 31

0



Value after reset: 0

Bit	Symbol	Function	R/W
0	RDFIE	Receive buffer full interrupt enabled 0: Interrupt request disabled by RDF = 1 1: Enable interrupt request by RDF = 1	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2

Note 1. This address is accessed from external debugger.

The JBICR register is used for controlling interrupt during JTAG boot.

### 2.8.6.4 CoreSight Component Registers

OCDREG has CoreSight Component Registers defined in ARM CoreSight Architecture. [Table 2.22](#) shows the list of the registers.

See [section 2.14. References \[7\]](#) for the detail of each register

**Table 2.22 CoreSight component registers**

Name	Address	Access size	R/W	Initial value
PID4	0x8001_1FD0	32 bits	Read-only	0x00000004
PID5	0x8001_1FD4	32 bits	Read-only	0x00000000
PID6	0x8001_1FD8	32 bits	Read-only	0x00000000
PID7	0x8001_1FDC	32 bits	Read-only	0x00000000
PID0	0x8001_1FE0	32 bits	Read-only	0x00000004
PID1	0x8001_1FE4	32 bits	Read-only	0x00000030
PID2	0x8001_1FE8	32 bits	Read-only	0x0000000A
PID3	0x8001_1FEC	32 bits	Read-only	0x00000000
CID0	0x8001_1FF0	32 bits	Read-only	0x0000000D
CID1	0x8001_1FF4	32 bits	Read-only	0x000000F0
CID2	0x8001_1FF8	32 bits	Read-only	0x00000005
CID3	0x8001_1FFC	32 bits	Read-only	0x000000B1

Note: S-TYPE-5, P-TYPE-2

## 2.9 CoreSight Cross Trigger Interface (CTI)

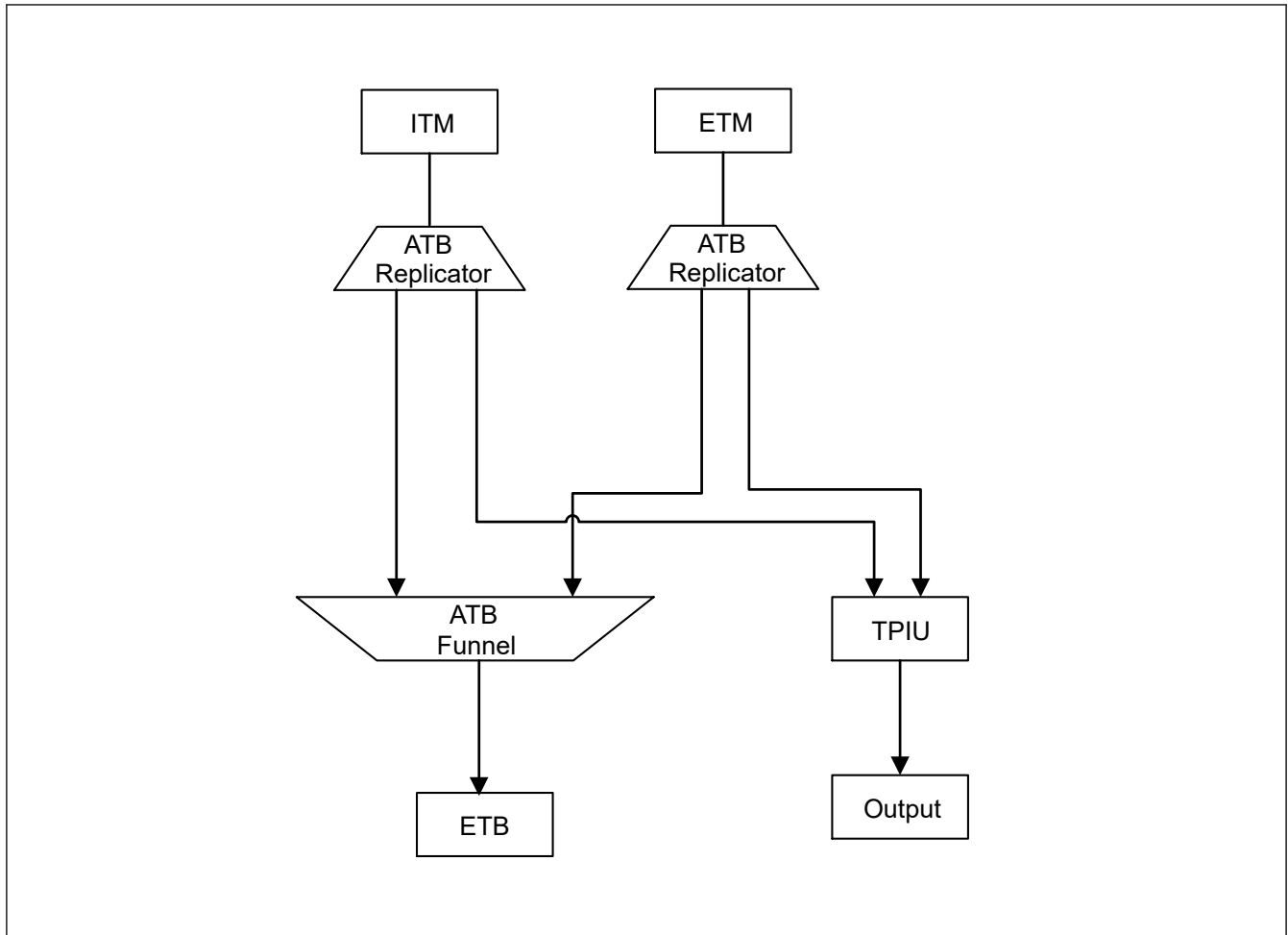
**Table 2.23** CTI channel

Number of CTI channel	CTITRIGIN		CTITRIGOUT	
	Channel No.	Signal	Channel No.	Signal
CTI	0	ACQCOMP	0	—
	1	FULL	1	—
	2	—	2	ETB FLUSHIN
	3	—	3	ETB TRIGIN
	4	—	4	—
	5	—	5	—
	6	—	6	—
	7	—	7	—
CTI0	0	Processor Halted	0	Processor debug request
	1	DWT Comparator Output 0	1	Processor Restart
	2	DWT Comparator Output 1	2	CTIIRQ[0] (connected to ICU0)
	3	DWT Comparator Output 2	3	CTIIRQ[1] (connected to ICU0)
	4	ETM Event Output 0	4	ETM Event Input 0
	5	ETM Event Output 1	5	ETM Event Input 1
	6	—	6	ETM Event Input 2
	7	—	7	ETM Event Input 3

The MCU supports processor CTI0 and CoreSight Common CTI which are implemented in Debug module. The ICU setting is necessary when CTIIRQ is used.

## 2.10 CoreSight ATB Funnel

The MCU has one CoreSight ATB funnel. The funnel has two ATB slaves and one ATB master, and it is used to select the debug trace sources from ETM and ITM to ETB. [Figure 2.3](#) shows the CoreSight ATB connection in the MCU.



**Figure 2.3 CoreSight ATB connection in the MCU**

Table 2.24 shows funnel ATB slave connection.

**Table 2.24 Funnel ATB slave connection.**

ATB slave number	Connected trace source
#0	CPU-ITM
#1	CPU-ETM

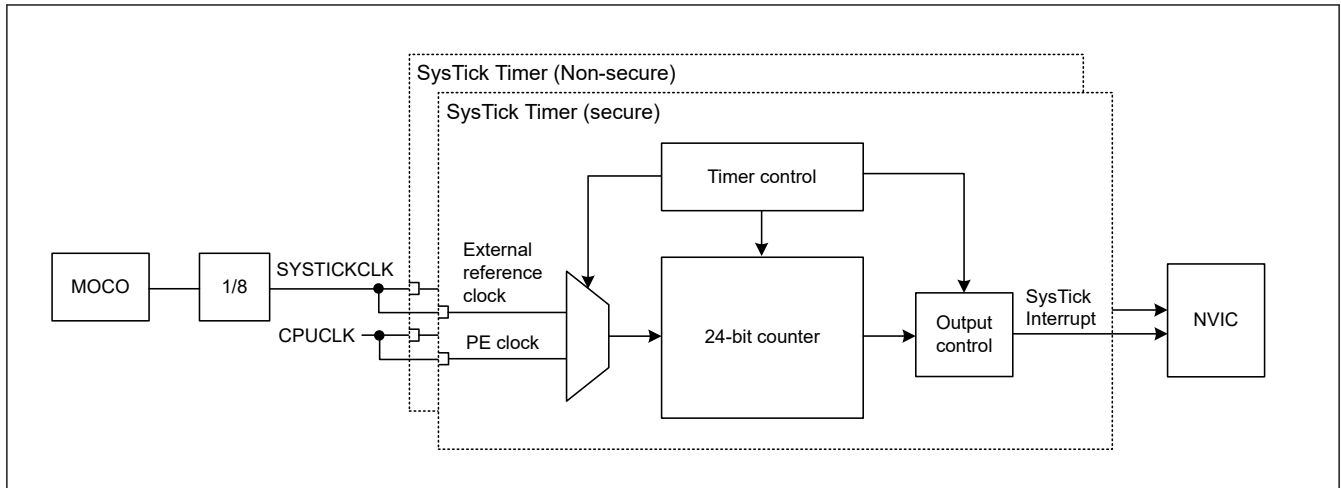
See section 2.14. References [5] for detail of ATB and Funnel.

## 2.11 SysTick System Timer

The MCU has two SysTick system timers, secure SysTick and non-secure SysTick.

The SysTick source is CPUCLK or SYSTICKCLK (MOCO/8). When the SysTick source is SYSTICKCLK (MOCO/8), the CPUCLK frequency must be 2 MHz or higher.

Figure 2.4 shows the SysTick Timer block diagram.



**Figure 2.4 SysTick Timer block diagram**

## 2.12 CoreSight Timestamp Generator

The MCU has a CoreSight Timestamp Generator to provide timestamp to ITM and ETM.

The timestamp is generated by a 64-bit counter operating with ICLK/2.

See [section 2.14. References \[5\]](#) for detail.

## 2.13 OCD Emulator Connection

In this product, the MCU confirms access permission for MCU resources by checking the current authentication level stored in OCDREG.

There are two ways to control authentication level:

- Through JTAG/SWD. See [Figure 2.5](#).
- Using software. See [Figure 2.6](#).



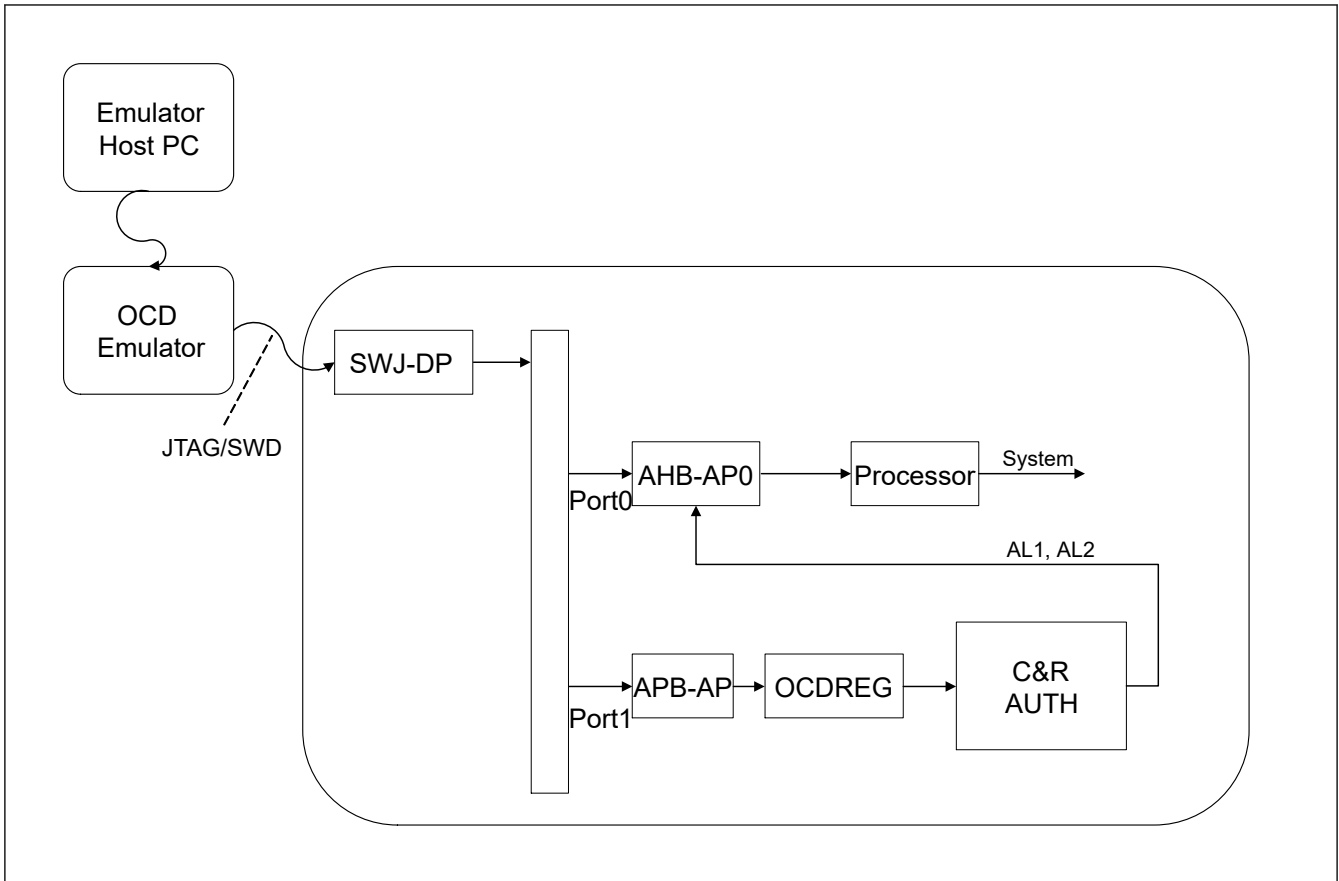


Figure 2.5 Control authentication level through JTAG/SWD

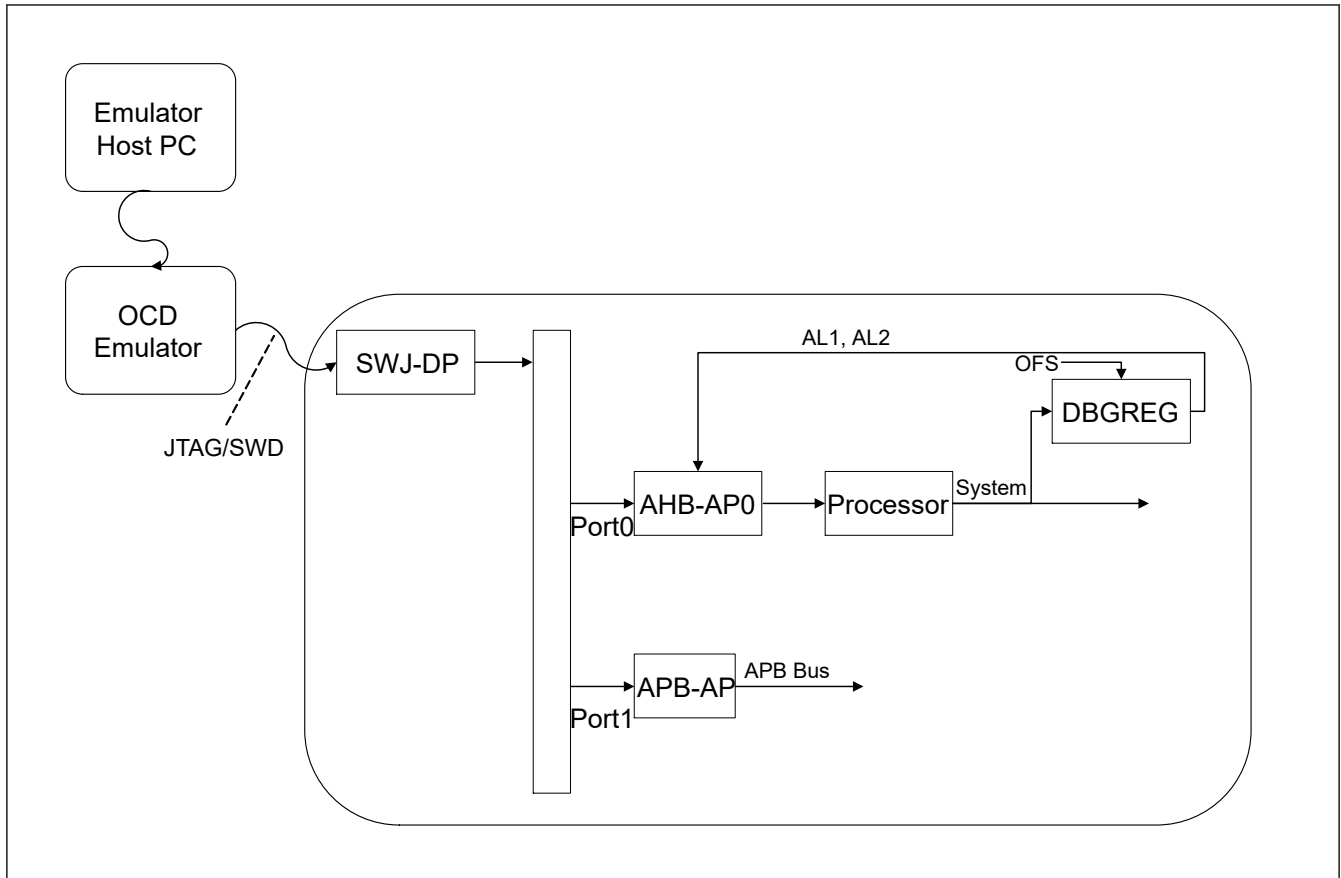


Figure 2.6 Control authentication level using software

### 2.13.1 SYOCD CR.DBGEN

See [section 2.7.3.1. Low power mode](#) for details.

### 2.13.2 Restriction in OCD Emulator Connecting

There are some restrictions regarding emulator access. This section describes those restrictions.

### 2.13.3 Starting Connection While in Low Power Mode

When starting a JTAG/SWD connection from an OCD emulator, the MCU must be in Normal, CPU Sleep, or CPU Deep Sleep mode. If the MCU is in Software Standby or Deep Software Standby mode, the OCD emulator can cause the MCU to hang.

### 2.13.4 Low Power Mode Change During Debugging

When the MCU is in OCD mode, the low power mode can be changed. However, system bus access from AHB-AP is prohibited in Deep Sleep, Software Standby, or Deep Software Standby mode. Only SWJ-DP, APB-AP, and OCDREG can be accessed from OCD emulator in these modes. [Table 2.25](#) shows the limitations.

Table 2.25 Low power mode change during debugging (1 of 2)

Current mode	Start OCD emulator connection	Change low power mode	Access AHB-AP and system bus	Access APB-AP and OCDREG
Normal	Available	Available	Available	Available
Sleep	Available	Available	Available	Available
Deep Sleep	Available	Available	N/A	Available
Software Standby	N/A	Available	N/A	Available

**Table 2.25 Low power mode change during debugging (2 of 2)**

Current mode	Start OCD emulator connection	Change low power mode	Access AHB-AP and system bus	Access APB-AP and OCDREG
Deep Software Standby	N/A	Available	N/A	Available

## 2.13.5 Connecting Sequence and JTAG/SWD Authentication

This section provides steps on how to authenticate the MCU. First, the section introduces which mechanism is used to authenticate the MCU. Next, using the previously described mechanism, the section describes steps for connecting sequence and authenticating the MCU using JTAG/SWD. See [section 2.8.5. DBGREG](#) and [section 2.8.6. OCDREG](#) for details about the registers control JTAG/SWD authentication.

### 2.13.5.1 JTAG/SWD Authentication Mechanism

The common platform of the MCU supports challenge-response authentication.

For challenge-response authentication, Boot FW authenticates.

### 2.13.5.2 Connecting Sequence and JTAG/SWD Authentication

For challenge-response authentication:

1. Connect the OCD debugger to the MCU through the JTAG or SWD interface.
2. Set up SWJ-DP to access DAP bus.  
In the setup, the OCD emulator must assert CDBGPWRUPREQ in the SWJDP Control Status Register, and then wait until CDBGPWRUPACK in the same register is asserted.
3. Set the APB-AP to access OCDREG. This APB-AP is connected to DAP bus port 1.
4. Set boot mode request.
5. Negate RES.
6. Check MCUSTAT.AL.  
If MCUSTAT.AL is higher than or equal to request AL, there is no need to authenticate.
7. Set authentication data.
8. Check MCUSTAT.AL.
  - (a) If MCUSTAT.AL does not request AL, authentication failed.
  - (b) If retry authentication: go to step 7.
9. Assert RES to move to single chip mode.
10. Set up the AHB-AP to access the system address space. The AHB-AP is connected to DAP bus port 0.
11. Start accessing the CPU debug resources using the AHB-AP.

## 2.14 References

- [1] Arm Limited., *Arm<sup>®</sup>v8-M Architecture Reference Manual* (ARM DDI 0553B)
- [2] Arm Limited., *Arm<sup>®</sup> Cortex<sup>®</sup>-M85 Processor Technical Reference Manual* (ARM 101924)
- [3] Arm Limited., *Arm<sup>®</sup> Cortex<sup>®</sup>-M85 Processor Integration and Implementation Manual* (ARM 101925)
- [4] Arm Limited., *Arm<sup>®</sup> Cortex<sup>®</sup>-M85 Processor Devices Generic User Guide* (ARM 101928)
- [5] Arm Limited., *Arm<sup>®</sup> CoreSight<sup>™</sup> SoC-400 Technical Reference Manual* (ARM DDI 0480G)
- [6] Arm Limited., *Arm<sup>®</sup> CoreSight<sup>™</sup> ETM-M85 Technical Reference Manual* (ARM 101926)
- [7] Arm Limited., *Arm<sup>®</sup> CoreSight<sup>™</sup> Architecture Specification* (ARM IHI 0029F)
- [8] Arm Limited., *Arm<sup>®</sup> Embedded Trace Macrocell Architecture Specification* (ARM IHI 0064H.a)
- [9] Arm Limited., *Arm<sup>®</sup> CoreSight SoC-400 User Guide* (ARM 100490)

## 2.15 Usage Notes

### 2.15.1 Restrictions for OSPI

There are restrictions to using OSPI, see [section 35.3.8.3. Memory Write Combination Mode](#) and [section 35.3.8.5. Restriction in 8D-8D-8D profile 1.0 format](#).

## 3. Operating Modes

### 3.1 Overview

[Table 3.1](#) shows the selection of operating modes by the mode-setting pin(MD pin) and JTAG accessible command. For details, see [section 3.3. Details of Operating Modes](#).

**Table 3.1 Selection of operating modes by the mode-setting pin**

Mode-setting pin (MD)	Operating mode	On-chip Flash
1	Single chip mode/ JTAG Boot Mode	Enable
0	SCI / USB boot mode	Enable

### 3.2 Operating Mode Types and Selection

[Table 3.1](#) show the relationship between levels on the mode-setting pins (MD) on release from the reset state and the operating mode selected at that time. For details on each of the operating modes, see [section 3.3. Details of Operating Modes](#). Operation starts with the on-chip Flash enabled regardless of the mode in which operation started.

### 3.3 Details of Operating Modes

#### 3.3.1 Single-Chip Mode

In single-chip mode, all I/O pins are available for use as input or output port, inputs or outputs for peripheral functions, or as interrupt inputs. When a reset is released while the MD pin is high, the MCU starts in single-chip mode and the on-chip flash is enabled.

#### 3.3.2 JTAG Boot Mode

In this mode, the on-chip flash memory programming routine (JTAG boot program), stored in the boot area within the MCU, is used. The on-chip flash, including code flash memory and data flash memory, can be modified from outside the MCU by using the JTAG and SWD interface. CSW.DbgSwEnable bit in APB-AP Control/Status Word register must be 1 in this mode.

To enter this mode, it is necessary to input the request from the JTAG and SWD-I/F during RES-pin reset.

#### 3.3.3 SCI Boot Mode

In this mode, the on-chip flash memory programming routine (SCI boot program), stored in a dedicated area within the MCU, is used. The on-chip flash including the code flash memory and data flash memory can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (UART) SCI. For details, see [section 49, Flash Memory](#). The MCU starts in SCI boot mode if the MD pin is held low on release from the reset state.

#### 3.3.4 USB Boot Mode

In this mode, the on-chip flash memory programming routine (USB boot program), stored in the boot area within the MCU, is used. The on-chip flash, including the code flash memory and data flash memory, can be modified from outside the MCU by using the USB. For details, see [section 49, Flash Memory](#). The MCU starts in USB boot mode if the MD pin is held low on release from the reset state.

### 3.4 Operating Modes Transitions

#### 3.4.1 Operation modes and the relationship of Mode Transition

[Figure 3.1](#) shows that the operation modes and the relationship of mode transition. Operating mode can be transitioned in the direction of the arrow at the figure.

JTAG Boot mode is not possible using POR.

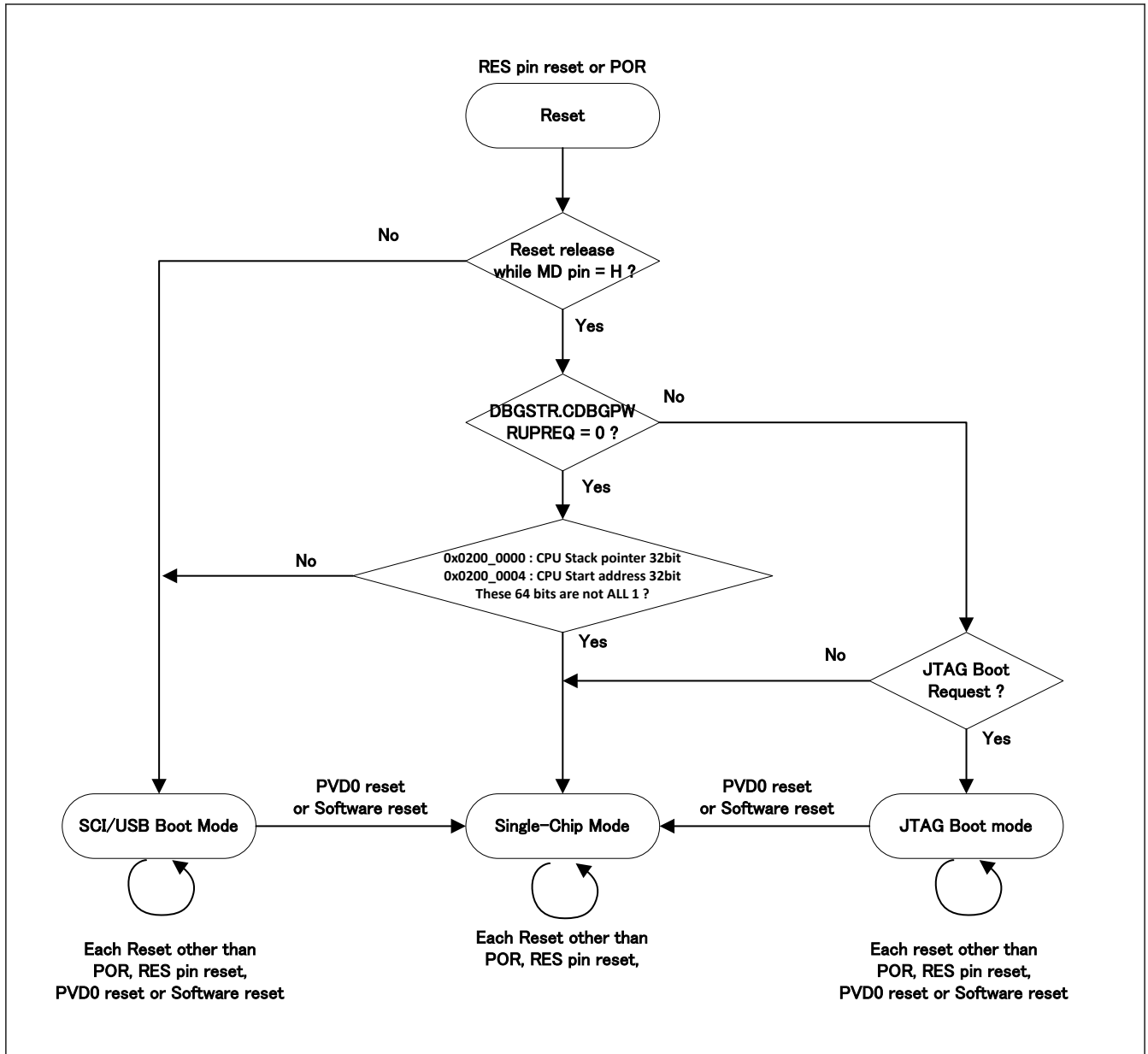


Figure 3.1 Operation modes and the relationship of mode transition

## 4. Address Space

### 4.1 Address Space

The MCU supports a 4 GB linear address space ranging from 0x0000\_0000 to 0xFFFF\_FFFF that can contain both program and data. In the address from 0x0000\_0000 to 0x5FFF\_FFFF, secure and non-secure region are isolated by using bit 28 of the address. For detail on the security attributes, see [section 40.3. Arm Security Features](#). [Figure 4.1](#) shows the memory map.

Address Map		IDAUMSAU Security_Attribution
0xFFFF_FFFF	Arm® Cortex®-M85	
0xE010_0000	Private peripheral bus	
0xE000_0000	Reserved area*2	
0xA000_0000	External address space (OSPI area)	
0x8000_0000	Reserved area*2	Non-secure
0x5050_0000	Peripheral I/O registers	
0x5020_0000	Reserved area*2	
0x5012_0000	Flash I/O registers	
0x5010_0000	Peripheral I/O registers	
0x5000_0000	Reserved area*2	
0x4050_0000	Peripheral I/O registers	Secure
0x4020_0000	Reserved area*2	
0x4012_0000	Flash I/O registers	
0x4010_0000	Peripheral I/O registers	
0x4000_0000	Reserved area*2	
0x3703_0400	On-chip flash (option-setting memory)	
0x3703_0050	Reserved area*2	
0x3700_3000	On-chip flash (data flash)	
0x3700_0000	Reserved area*2	
0x3600_0400	Standby SRAM	Non-secure
0x3600_0000	Reserved area*2	
0x320E_0000	On-chip SRAM	
0x3206_0000	Reserved area*2	
0x3000_4000	DTCM	
0x3000_0000	Reserved area*2	
0x2703_0400	On-chip flash (option-setting memory)	
0x2703_0050	Reserved area*2	
0x2700_3000	On-chip flash (data flash)	Non-secure callable for CPU
0x2700_0000	Reserved area*2	
0x2600_0400	Standby SRAM	
0x2600_0000	Reserved area*2	
0x220E_0000	On-chip SRAM	Secure for other bus masters
0x2206_0000	Reserved area*2	
0x2000_4000	DTCM	
0x2000_0000	Reserved area*2	
0x1300_A300	On-chip flash (option-setting memory)	
0x1300_A100	Reserved area*2	
0x1300_81B4	On-chip flash (Factory Flash)	
0x1300_80F0	Reserved area*2	
0x1228_0000	On-chip flash (code flash) (read only)*1	Non-secure
0x1200_0000	Reserved area*2	
0x1000_4000	ITCM	
0x1000_0000	Reserved area*2	
0x0300_A300	On-chip flash (option-setting memory)	
0x0300_A100	Reserved area*2	
0x0300_81B4	On-chip flash (Factory Flash)	
0x0300_80F0	Reserved area*2	
0x0228_0000	On-chip flash (code flash) (read only)*1	Secure for other bus masters
0x0200_0000	Reserved area*2	
0x0000_4000	ITCM	
0x0000_0000	ITCM	

Note 1. See Table 4.1 and Table 4.2. The capacity of the flash differs depending on the product.  
 Note 2. Do not access reserved areas.

Figure 4.1 Memory map



**Table 4.1 Capacity of the code flash memory, data flash memory, and On-chip SRAM in Secure alias**

Code flash memory			Data flash memory		On-chip SRAM	
Capacity	Address		Capacity	Address	Capacity	Address
	Linear mode	Dual mode (BANKSEL.BANKSWP[2:0] = 111b)				
1 MB	0x0200_0000 to 0x020F_FFFF	upper side bank 0x0220_0000 to 0x0227_FFFF	12 KB	0x2700_0000 to 0x2700_2FFF	512 KB	0x2206_0000 to 0x220D_FFFF
		lower side bank 0x0200_0000 to 0x0207_FFFF				

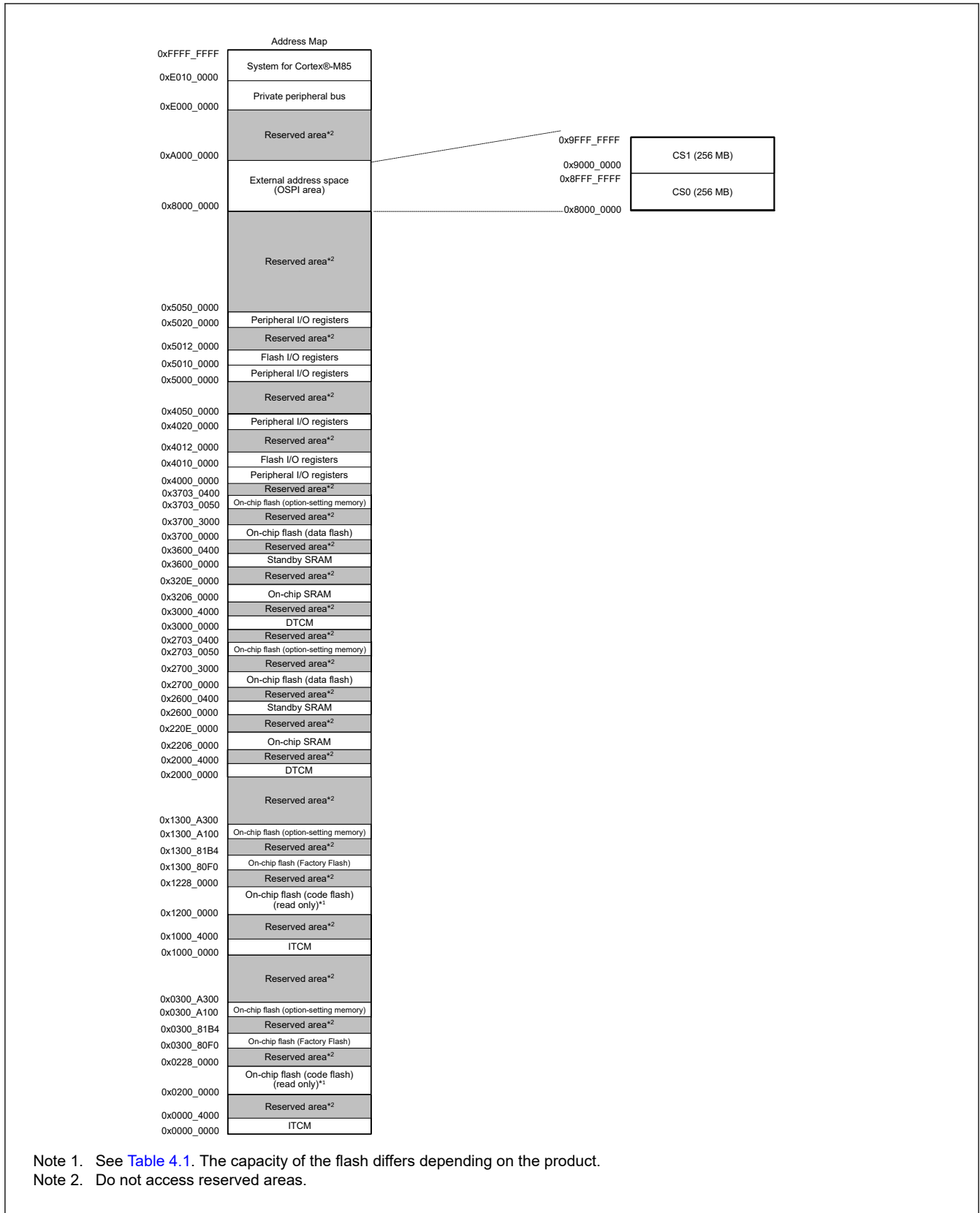
**Table 4.2 Capacity of the code flash memory, data flash memory, and On-chip SRAM in Non-secure alias**

Code flash memory			Data flash memory		On-chip SRAM	
Capacity	Address		Capacity	Address	Capacity	Address
	Linear mode	Dual mode (BANKSEL.BANKSWP[2:0] = 111b)				
1 MB	0x1200_0000 to 0x120F_FFFF	upper side bank 0x1220_0000 to 0x1227_FFFF	12 KB	0x3700_0000 to 0x3700_2FFF	512 KB	0x3206_0000 to 0x320D_FFFF
		lower side bank 0x1200_0000 to 0x1207_FFFF				

## 4.2 External Address Space

The external address space is divided into OSPI areas (CS0 and CS1). The two OSPI areas (CS0 and CS1) each correspond to the OM\_CS<sub>n</sub> signal output from a OM\_CS<sub>n</sub> (n = 0, 1) pin.

Figure 4.2 shows the address ranges associated with the individual OSPI areas (CS0 and CS1).



Note 1. See Table 4.1. The capacity of the flash differs depending on the product.  
 Note 2. Do not access reserved areas.

Figure 4.2 Detailed address map for CS area and OSPI area

## 5. Resets

### 5.1 Overview

This MCU provides the following 13 types of reset.

It also has a function to control the reset generation by only Secure program access.

System reset is all factors such that can initialize CPU core. All reset factors except VBATT\_POR are system reset for this product.

On the other hand, CPU has a reset that initializes itself individually. That reset is not a system reset. See [section 2, CPU](#) for details.

[Table 5.1](#) lists the reset names and sources.

**Table 5.1 Reset names and sources**

Reset Name	Source	
System reset	RES Pin reset	The RES pin voltage input is driven low
	Power-on reset (POR)	VCC falls (voltage detection: $V_{POR}$ ) <sup>*1</sup>
	Voltage Monitor 0 reset	VCC falls (voltage detection: $V_{det0}$ ) <sup>*1</sup>
	Voltage Monitor 1 reset	VCC falls (voltage detection: $V_{det1}$ ) <sup>*1</sup>
	Voltage Monitor 2 reset	VCC falls (voltage detection: $V_{det2}$ ) <sup>*1</sup>
	Independent Watchdog Timer reset	IWDT underflow or refresh error
	Watchdog Timer reset	WDT underflow or refresh error for CPU
	CPU Lockup reset	This reset is generated when CPU encounters lockup
	Bus Error reset	BUS error (MSAU error, MMPU error, Illegal address error, STZF error, Slave Bus error, Bufferable write error)
	Common Memory Error reset	RAM error (Parity error of SRAM and Standby SRAM)
	Deep Software Standby reset	Deep Software Standby mode is canceled by an interrupt
	Software reset	Register setting (use the software reset bit AIRCR.SYSRESETREQ)
VBATT_POR reset	VBATT_R falls (voltage detection: $V_{PDR(BATR)}$ ) <sup>*1</sup>	

Note 1. For details on the voltages to be monitored ( $V_{POR}$ ,  $V_{det0}$ ,  $V_{det1}$ ,  $V_{det2}$ , and  $V_{PDR(BATR)}$ ), see [section 7, Programmable Voltage Detection \(PVD\)](#), [section 11, Battery Backup Function](#), and [section 52, Electrical Characteristics](#).

The internal state and pins are initialized by a reset. [Table 5.2](#) and [Table 5.3](#) list the targets initialized by resets.

Some of these registers are initialized when returning from Software Standby mode. See [section 10, Low Power Modes](#).

In this table, registers of modules marked as "Undesfined" in Software Standby mode are initialized.

**Table 5.2 Reset detect flags initialized by each reset source (1 of 4)**

Flag to be initialized	Reset source						
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	CPU Lockup reset	Voltage monitor 1 reset
Power-On Reset Detect Flag (RSTSR0.PORF)	✓	—	—	—	—	—	—
Voltage Monitor 0 Reset Detect Flag (RSTSR0.PVD0RF)	✓	✓	—	—	—	—	—
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	✓	✓	✓	—	—	—	—
Watchdog Timer Reset Detect Flag (RSTSR1.WDT0RF)	✓	✓	✓	—	—	—	—
CPU Lockup Reset Detect Flag (RSTSR1.CLU0RF)	✓	✓	✓	—	—	—	—

**Table 5.2 Reset detect flags initialized by each reset source (2 of 4)**

Flag to be initialized	Reset source						
	RES pin reset	Power-on reset	Voltage monitor 0 reset	Independent watchdog timer reset	Watchdog timer reset	CPU Lockup reset	Voltage monitor 1 reset
Voltage Monitor 1 Reset Detect Flag (RSTSR0.PVD1RF)	✓	✓	✓	—	—	—	—
Voltage Monitor 2 Reset Detect Flag (RSTSR0.PVD2RF)	✓	✓	✓	—	—	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	✓	✓	✓	—	—	—	—
Bus error Reset Detect Flag (RSTSR1.BUSRF)	✓	✓	✓	—	—	—	—
Common Memory Error Reset Detect Flag (RSTSR1.CMRF)	✓	✓	✓	—	—	—	—
Deep Software Standby Reset Detect Flag (RSTSR0.DPSRSTF)	✓	✓	✓	—	—	—	—
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	✓	—	—	—	—	—

**Table 5.2 Reset detect flags initialized by each reset source (3 of 4)**

Flag to be initialized	Reset source							
	Voltage monitor 2 reset	Software reset	Bus error reset	Common memory error reset	Deep software standby reset			VBATT_POR*1
					Deep Software Standby mode 1	Deep Software Standby mode 2	Deep Software Standby mode 3	
Power-On Reset Detect Flag (RSTSR0.PORF)	—	—	—	—	—	—	—	—
Voltage Monitor 0 Reset Detect Flag (RSTSR0.PVD0RF)	—	—	—	—	—	—	—	—
Independent Watchdog Timer Reset Detect Flag (RSTSR1.IWDTRF)	—	—	—	—	—	✓	✓	—
Watchdog Timer Reset Detect Flag (RSTSR1.WDT0RF)	—	—	—	—	—	✓	✓	—
CPU Lockup Reset Detect Flag (RSTSR1.CLU0RF)	—	—	—	—	—	✓	✓	—
Voltage Monitor 1 Reset Detect Flag (RSTSR0.PVD1RF)	—	—	—	—	—	—	—	—
Voltage Monitor 2 Reset Detect Flag (RSTSR0.PVD2RF)	—	—	—	—	—	—	—	—
Software Reset Detect Flag (RSTSR1.SWRF)	—	—	—	—	—	✓	✓	—
Bus error Reset Detect Flag (RSTSR1.BUSRF)	—	—	—	—	—	✓	✓	—
Common Memory Error Reset Detect Flag (RSTSR1.CMRF)	—	—	—	—	—	✓	✓	—

**Table 5.2 Reset detect flags initialized by each reset source (4 of 4)**

Flag to be initialized	Reset source							
	Voltage monitor 2 reset	Software reset	Bus error reset	Common memory error reset	Deep software standby reset			VBATT_POR*1
					Deep Software Standby mode 1	Deep Software Standby mode 2	Deep Software Standby mode 3	
Deep Software Standby Reset Detect Flag (RSTSR0.DPSRSTF)	—	—	—	—	—	—	—	—
Cold Start/Warm Start Determination Flag (RSTSR2.CWSF)	—	—	—	—	—	—	—	—

Note: ✓ : Initialized to 0  
 — : Not initialized

Note 1. For VBATT\_POR details, See [section 11, Battery Backup Function](#).

**Table 5.3 Module-related registers initialized by each reset source (1 of 4)**

Registers to be initialized		Reset source							
		RES# pin reset	Power-on reset	Voltage monitoring 0 reset	Independent watchdog timer reset	Watchdog timer reset	CPU Lockup reset	Voltage monitoring 1 reset	
Voltage Monitor Function 1 registers	PVD1CR0, PVD1CMPCR, PVD1FCR	✓	✓	✓	✓	✓	—	—	
	PVD1CR1, PVD1SR	✓	✓	✓	✓	✓	—	—	
Voltage Monitor Function 2 registers	PVD2CR0, PVD2CMPCR, PVD2FCR	✓	✓	✓	✓	✓	—	—	
	PVD2CR1, PVD2SR	✓	✓	✓	✓	✓	—	—	
SOSC registers	SOSCCR, SOMCR	—	—	—	—	—	—	—	
LOCO registers	LOCOUTCR	—	✓	✓	—	—	—	—	
MOSC registers	MOMCR	✓	✓	✓	✓	✓	✓	✓	
HOCO control register 2	HOCO CR2	—	✓	✓	—	—	—	—	
Pin states (except XCIN / XCOU)		✓	✓	✓	✓	✓	✓	✓	
Pin states (XCIN / XCOU)		—	—	—	—	—	—	—	
IO capture and tamper detection such as VBAT (sampling timing for RTC) (RTCIC0-2)	VBTICTLR, VBTICTLR2, VBTADSR, VBTADCR1, VBTADCR2	—	—	—	—	—	—	—	
VBATT Battery power supply switch control register 1	VBTBPCR1	✓	✓	✓	✓	✓	✓	✓	
VBATT Battery Power Supply Switch Control register 2	VBTBPCR2	—	—	—	—	—	—	—	
VBATT Backup Enable register	VBTB ER	—	✓	—	—	—	—	—	

**Table 5.3 Module-related registers initialized by each reset source (2 of 4)**

Registers to be initialized		Reset source						
		RES# pin reset	Power-on reset	Voltage monitoring 0 reset	Independent watchdog timer reset	Watchdog timer reset	CPU Lockup reset	Voltage monitoring 1 reset
Battery Backup register	VBTBKR[n]	—	—	—	—	—	—	—
VBATT Input Monitor Register	VBTIMONR	—	—	—	—	—	—	—
Independent Watchdog Timer registers	IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR	✓	✓	✓	✓	✓	✓	✓
Realtime Clock register		—	—	—	—	—	—	—
Ultra-low-power timer registers	ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, ULPTCMSR	✓	✓	✓	✓	✓	✓	✓
USBFS registers	DPUSR0R, DPUSR1R	✓	✓	✓	✓	✓	✓	✓
Reset Flag	BUSnERRADD (n = 4, 5) BUSnERRRW (n = 4, 5) BMSAnERRADD (n = 4, 5) BMSAnERRRW (n = 4, 5) BUSnERRSTAT (n = 1 to 5, 10) MBWERRSTAT, SBWERRSTAT, SRAMESR, SRAMEAR2, STBRAMEAR	✓	✓	✓	✓	✓	✓	✓
Reset Flag	See <a href="#">Table 5.2</a>							
Low Power Function registers	DPSBYCR, DPSWCR, DPSIER0 to DPSIER3, DPSIFR0 to DPSIFR3, DPSIEGR0 to DPSIEGR2 LPSCR FWEPROR SSCR1 LVOCR	✓	✓	✓	✓	✓	✓	✓
Low power Function registers	SYOCDRCR	—	✓	✓	—	—	—	—
System Reset Mask Control Register *7	SYRSTMSK0, SYRSTMSK2	✓	✓	✓	—	—	—	—
ARM Debug function	MCUSTAT, MCUCTRL, JBMADR, DBGSTR, DBGSTOPCR, DBGAUTH0, DBGAUTH1, CACHEDBGCR, TRPORTCR	—	✓	✓	—	—	—	—
TRCLK Control registers	TRCKCR	—	✓	✓	—	—	—	—
Power Gating Control registers	PDRAMSCR0, PDRAMSCR1	—	✓	✓	—	—	—	—
other than specified		✓	✓	✓	✓	✓	✓	✓

Table 5.3 Module-related registers initialized by each reset source (3 of 4)

Registers to be initialized		Reset source							
		Voltage monitoring 2 reset	Software reset	Bus error reset	Common memory error reset	Deep software standby reset			VBATT-selected voltage power-on reset
						Deep Software Standby mode 1 reset	Deep Software Standby mode 2 reset	Deep Software Standby mode 3 reset	
Voltage Monitor Function 1 registers	PVD1CR0, PVD1CMPCR, PVD1FCR	—	—	—	—	—	—	—	—
	PVD1CR1, PVD1SR	—	—	—	—	✓	✓	✓	—
Voltage Monitor Function 2 registers	PVD2CR0, PVD2CMPCR, PVD2FCR	—	—	—	—	—	—	—	—
	PVD2CR1, PVD2SR	—	—	—	—	✓	✓	✓	—
SOSC registers	SOSCCR, SOMCR	—	—	—	—	—	—	—	✓
LOCO registers	LOCOUTCR	—	—	—	—	—	✓	✓	—
MOSC registers	MOMCR	✓	✓	✓	✓	—	—	—	—
HOCO control register 2	HOCOCR2	—	—	—	—	✓*3	✓*3	✓*3	—
Pin states (except XCIN / XCOUT)		✓	✓	✓	✓	✓*2	✓*2	✓*2	—
Pin states (XCIN / XCOUT)		—	—	—	—	—	—	—	✓
IO capture and tamper detection such as VBAT (sampling timing for RTC) (RTCIC0-2)	VBTICTLR, VBTICTLR2, VBTADSR, VBTADCR1, VBTADCR2	—	—	—	—	—	—	—	✓
VBATT Battery Power Supply Switch Control register 1	VBTBPCR1	✓	✓	✓	✓	—	—	—	—
VBATT Battery Power Supply Switch Control register 2	VBTBPCR2	—	—	—	—	—	—	—	✓
VBATT Backup Enable register	VBTBER	—	—	—	—	—	—	—	—
Battery Backup register	VBTBKR[n]	—	—	—	—	—	—	—	✓
VBATT Input Monitor Register	VBTIMONR	—	—	—	—	—	—	—	—
Independent Watchdog Timer registers	IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR	✓	✓	✓	✓	—	✓	✓	—
Realtime Clock*1 register		—	—	—	—	—	—	—	—

**Table 5.3 Module-related registers initialized by each reset source (4 of 4)**

Registers to be initialized		Reset source							
		Voltage monitoring 2 reset	Software reset	Bus error reset	Common memory error reset	Deep software standby reset			VBATT-selected voltage power-on reset
						Deep Software Standby mode 1 reset	Deep Software Standby mode 2 reset	Deep Software Standby mode 3 reset	
Ultra-low-power timer registers	ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, ULPTCMSR	✓	✓	✓	✓	—	✓	✓	—
USBFS registers	DPUSR0R, DPUSR1R	✓	✓	✓	✓	—	✓	✓	—
Reset Flag	BUSnERRADD (n = 4, 5) BUSnERRRW (n = 4, 5) BMSAnERRADD (n = 4, 5) BMSAnERRRW (n = 4, 5) BUSnERRSTAT (n = 1 to 5, 10) MBWERRSTAT, SBWERRSTAT, SRAMESR, SRAMEAR2, STBRAMEAR	✓	✓	—	—	✓	✓	✓	—
Reset Flag	See <a href="#">Table 5.2</a>								
Low Power Function registers	DPSBYCR, DPSWCR, DPSIER0 to DPSIER3, DPSIFR0 to DPSIFR3, DPSIEGR0 to DPSIEGR2 LPSCR FWEPROR SSCR1 LVOCR	✓	✓	✓	✓	—	—	—	—
Low Power Function registers	SYOCDRCR	—	—	—	—	—	—	—	—
System Reset Mask Control Register *7	SYRSTMSK0, SYRSTMSK2	—	—	—	—	—	—	—	—
ARM Debug function	MCUSTAT, MCUCTRL, JBMDR, DBGSTR, DBGSTOPCR, DBGAUTH0, DBGAUTH1, CACHEDBGCR, TRPORTCR	—	—	—	—	✓*3	✓*3	✓*3	—
TRCLK Control registers	TRCKCR	—	—	—	—	✓*3	✓*3	✓*3	—
Power Gating Control registers	PDRAMSCR0, PDRAMSCR1	—	—	—	—	✓	✓	✓	—
Other than specified		✓	✓	✓	✓	✓	✓	✓	—

Note: ✓ : Initialized  
— : Not initialized

Note 1. RTC has Software reset for RTC. Some control bits are not initialized by all types of resets. For details on the target bits, [section 24, Realtime Clock \(RTC\)](#).

Note 2. Depend on setting of DPSBYCR.IOKEEP.

Note 3. After the debugger authentication passes, setting SYOCDRCR.DBGEN to 1 does not initialize this register or the function. For details, see [section 2, CPU](#).

SOSC and LOCO can be selected as the clock sources of the RTC.

[Table 5.4](#) and [Table 5.5](#) show the states of SOSC and LOCO when a reset occurs.



**Table 5.4 States of SOSC when a reset occurs**

		Reset source	
		VBATT_POR reset	Other
SOSC	Enable or disable	Initialized to enable	Continue with the state that was selected before the reset occurred
	Drive capability	Initialized to standard drive capability	

**Table 5.5 States of LOCO when a reset occurs**

		Reset source	
		Power-on reset, Voltage-Monitoring0 reset, Deep software standby 2,3 reset	Other
LOCO	Enable or disable	Initialized to enable	Continue with the accuracy that was trimmed by LOCOUTCR
	Oscillation accuracy*1	Initialized to accuracy before trimming by LOCOUTCR (accuracy: ± 15%)	

Note 1. If the LOCO that is trimmed by LOCOUTCR is selected as the RTC source clock, please be careful that LOCO oscillation accuracy will be initialized by Power-on reset, Voltage-Monitor 0 reset and Deep software standby 2, 3 reset.

When a reset is released, reset exception handling starts. For details on the reset exception handling, see [section 5.3.12. Determination of Reset Generation Source](#).

[Table 5.6](#) lists the pin related to the reset function.

For details on the reset exception handling, see [section 5.3.12. Determination of Reset Generation Source](#).

**Table 5.6 Pin related to reset**

Pin name	I/O	Function
RES	Input	Reset pin

## 5.2 Register Descriptions

### 5.2.1 RSTSAR : Reset Security Attribution Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x3C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	NONS EC2	NONS EC1	NONS EC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non-secure Attribute bit 0 Target register: Reset Status Register 0 0: Secure 1: Non-secure	R/W
1	NONSEC1	Non-secure Attribute bit 1 Target register: Reset Status Register 1 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
2	NONSEC2	Non-secure Attribute bit 2 Target register: Reset Status Register 2 0: Secure 1: Non-secure	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

#### NONSEC0 bit (Non-secure Attribute bit 0)

This bit controls the security attribute of RSTSR0.

#### NONSEC1 bit (Non-secure Attribute bit 1)

This bit controls the security attribute of RSTSR1.

#### NONSEC2 bit (Non-secure Attribute bit 2)

This bit controls the security attribute of RSTSR2.

### 5.2.2 RSTSR0 : Reset Status Register 0

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA40

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DPSR STF	—	—	—	PVD2 RF	PVD1 RF	PVD0 RF	PORF
Value after reset:	x <sup>1</sup>	0	0	0	x <sup>1</sup>	x <sup>1</sup>	x <sup>1</sup>	x <sup>1</sup>

Bit	Symbol	Function	R/W
0	PORF	Power-On Reset Detect Flag 0: Power-on reset not detected 1: Power-on reset detected	R/W <sup>2</sup>
1	PVD0RF	Voltage Monitor 0 Reset Detect Flag 0: Voltage monitor 0 reset not detected 1: Voltage monitor 0 reset detected	R/W <sup>2</sup>
2	PVD1RF	Voltage Monitor 1 Reset Detect Flag 0: Voltage monitor 1 reset not detected 1: Voltage monitor 1 reset detected	R/W <sup>2</sup>
3	PVD2RF	Voltage Monitor 2 Reset Detect Flag 0: Voltage monitor 2 reset not detected 1: Voltage monitor 2 reset detected	R/W <sup>2</sup>
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	DPSRSTF	Deep Software Standby Reset Flag 0: Deep Software Standby mode cancellation not requested by an interrupt or a reset <sup>*3</sup> . 1: Deep Software Standby mode cancellation requested by an interrupt or a reset <sup>*3</sup> .	R/W <sup>2</sup>

Note: S-TYPE-3, P-TYPE-2

Note 1. The value after reset depends on the reset source.

Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

Note 3. Independent watchdog timer reset, Voltage Monitor 1 reset, Voltage Monitor 2 reset.

#### PORF flag (Power-On Reset Detect Flag)

The PORF flag indicates that a power-on reset occurred.

[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When a reset listed in [Table 5.2](#) occurs
- When 0 is written after 1 is read from PORF flag.

#### **PVD0RF flag (Voltage Monitor 0 Reset Detect Flag)**

The PVD0RF flag indicates that the Voltage Monitor 0 reset has occurred.

[Setting condition]

- When a voltage monitor 0 reset occurs.

[Clearing conditions]

- When a reset listed in [Table 5.2](#) occurs
- When 0 is written after 1 is read from PVD0RF flag.

#### **PVD1RF flag (Voltage Monitor 1 Reset Detect Flag)**

The PVD1RF flag indicates that the Voltage Monitor 1 reset has occurred.

[Setting condition]

- When a voltage monitor 1 reset occurs.

[Clearing conditions]

- When a reset listed in [Table 5.2](#) occurs
- When 0 is written after 1 is read from PVD1RF flag.

#### **PVD2RF flag (Voltage Monitor 2 Reset Detect Flag)**

The PVD2RF flag indicates that the Voltage Monitor 2 reset has occurred.

[Setting condition]

- When a voltage monitor 2 reset occurs.

[Clearing conditions]

- When a reset listed in [Table 5.2](#) occurs
- When 0 is written after 1 is read from PVD2RF flag.

#### **DPSRSTF flag (Deep Software Standby Reset Flag)**

The DPSRSTF flag indicates that Deep Software Standby mode is canceled by an external or internal interrupt, and that an internal reset (Deep Software Standby reset) occurs when an exception from Deep Software Standby mode occurs.

[Setting condition]

- When Deep Software Standby mode is cancelled by an external or an internal interrupt. For details, see [section 10, Low Power Modes](#).

[Clearing conditions]

- When a reset listed in [Table 5.2](#) occurs.
- When 0 is written after 1 is read from DPSRSTF flag.

### 5.2.3 RSTSR1 : Reset Status Register 1

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x0C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CMRF	—	—	—	BUSRF	—	—	—	—	—	CLU0RF	—	SWRF	WDT0RF	IWDTRF
Value after reset:	0	x*1	0	0	0	x*1	0	0	0	0	0	x*1	0	x*1	x*1	x*1

Bit	Symbol	Function	R/W
0	IWDTRF	Independent Watchdog Timer Reset Detect Flag 0: Independent watchdog timer reset not detected 1: Independent watchdog timer reset detected	R/W <sup>2</sup>
1	WDT0RF	Watchdog Timer Reset Detect Flag 0: Watchdog timer reset not detected 1: Watchdog timer reset detected	R/W <sup>2</sup>
2	SWRF	Software Reset Detect Flag 0: Software reset not detected 1: Software reset detected	R/W <sup>2</sup>
3	—	This bit is read as 0. The write value should be 0.	
4	CLU0RF	CPU Lockup Reset Detect Flag 0: CPU Lockup reset not detected 1: CPU Lockup reset detected	R/W <sup>2</sup>
9:5	—	These bits are read as 0. The write value should be 0.	R/W
10	BUSRF	Bus Error Reset Detect Flag 0: Bus error reset not detected 1: Bus error reset detected	R/W <sup>2</sup>
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	CMRF	Common Memory Error Reset Detect Flag 0: Common memory error reset not detected 1: Common memory error reset detected	R/W <sup>2</sup>
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note 1. The value after reset depends on the reset source.

Note 2. Only 0 can be written to clear the flag. The flag must be cleared by writing 0 after 1 is read.

#### IWDTRF flag (Independent Watchdog Timer Reset Detect Flag)

The IWDTRF flag indicates that an Independent watchdog timer reset occurred.

[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in [Table 5.2](#) occurs
- When 0 is written after 1 is read from IWDTRF flag.

#### WDT0RF flag (Watchdog Timer Reset Detect Flag)

The WDT0RF flag indicates that a Watchdog timer reset occurred.

[Setting condition]

- When a watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in [Table 5.2](#) occurs
- When 0 is written after 1 is read from WDTORF flag.

#### **SWRF flag (Software Reset Detect Flag)**

The SWRF flag indicates that a Software reset occurred.

[Setting condition]

- When a software reset occurs.

[Clearing conditions]

- When a reset listed in [Table 5.2](#) occurs
- When 0 is written after 1 is read from SWRF flag.

#### **CLU0RF flag (CPU Lockup Reset Detect Flag)**

The CLU0RF flag indicates that a CPU Lockup reset occurred.

[Setting condition]

- When a CPU Lockup reset occurs

[Clearing conditions]

- When a reset listed in [Table 5.2](#) occurs
- When 0 is written after 1 is read from CLU0RF flag.

#### **BUSRF flag (Bus Error Reset Detect Flag)**

The BUSRF flag indicates that BUS error reset (MSAU error, MMPU error, Illegal address error, Slave TrustZone Filter error, Slave Bus error, Bufferable write error) occurred.

[Setting condition]

- When a bus error reset occurs.

[Clearing conditions]

- When a reset listed in [Table 5.2](#) occurs
- When 0 is written after 1 is read from BUSRF flag.

#### **CMRF flag (Common Memory Error Reset Detect Flag)**

The CMRF flag indicates that a Common memory error reset (Parity error of SRAM and Standby SRAM) occurred.

[Setting condition]

- When a Common memory error reset occurs.

[Clearing conditions]

- When a reset listed in [Table 5.2](#) occurs
- When 0 is written after 1 is read from CMRF flag.

There are several causes for setting this reset flag.

If you want to identify the reset factor, see [section 13, Interrupt Controller Unit \(ICU\)](#).

### 5.2.4 RSTSR2 : Reset Status Register 2

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA44

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CWSF
Value after reset:	0	0	0	0	0	0	0	x <sup>*1</sup>

Bit	Symbol	Function	R/W
0	CWSF	Cold/Warm Start Determination Flag 0: Cold start 1: Warm start	R/W <sup>2</sup>
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note 1. The value after reset depends on the reset source.

Note 2. Only 1 can be written to set the flag.

RSTSR2 determines whether a power-on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start).

#### CWSF flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing, either cold start or warm start. CWSF flag is initialized by a power-on reset. It is not initialized by a reset signal generated by the RES pin.

[Setting condition]

- When 1 is written by software. Writing 0 to CWSF does not set it to 0.

[Clearing condition]

- When a reset listed in [Table 5.2](#) occurs.

### 5.2.5 SYRSTMSK0 : System Reset Mask Control Register 0

Base address: SYSC = 0x4001\_E000

Offset address: 0xAD0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BUSM ASK	CMMA SK	—	CLU0 MASK	—	SWMA SK	WDT0 MASK	IWDT MASK
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTMASK	Independent Watchdog Timer Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
1	WDT0MASK	Watchdog Timer Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
2	SWMASK	Software Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	CLU0MASK	CPU Lockup Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W

Bit	Symbol	Function	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	CMMASK	Common Memory Error Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
7	BUSMASK	Bus Error Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W

Note: S-TYPE-6, P-TYPE-2

Note: Set the PRCR.PRC5 bit to 1 (write enabled) before rewriting this register.

The SYRSTMSK0 is a register that controls the occurrence of reset.

The IWDTMASK bit cannot be rewritten while the independent watchdog timer is operating. The WDT0MASK bit cannot be rewritten while the watchdog timer is operating.

### 5.2.6 SYRSTMSK2 : System Reset Mask Control Register 2

Base address: SYSC = 0x4001\_E000

Offset address: 0xAD8

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PVD2 MASK	PVD1 MASK
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PVD1MASK	Voltage Monitor 1 Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
1	PVD2MASK	Voltage Monitor 2 Reset Mask 0: Reset occurrence is enabled 1: Reset occurrence is disabled	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2

Note: Set the PRCR.PRC5 bit to 1 (write enabled) before rewriting this register.

The SYRSTMSK2 is a register that controls the occurrence of reset.

## 5.3 Operation

### 5.3.1 RES Pin Reset

The RES pin generates this reset. When the RES pin is driven low, all the processing in progress is aborted and the MCU enters a reset state. To successfully reset the MCU, the RES pin must be held low for the power supply stabilization time specified at power-on.

When the RES pin is driven high from low, the internal reset is canceled after the post-RES cancellation wait time ( $t_{RESWT}$ ) elapses. The CPU then starts the reset exception handling.

For details, see [section 52, Electrical Characteristics](#).

### 5.3.2 Power-On Reset

The power-on reset circuit generates this internal reset. If the RES pin is in a high-level state when power is supplied, a power-on reset is generated.

After VCC exceeds  $V_{POR}$  and the specified period (power-on reset time) elapses, the internal reset is canceled, and the CPU starts the reset exception handling. The power-on reset time is the stabilization period for the external power supply and the MCU circuit.

After a power-on reset is generated, the PORF flag in the RSTSR0 is set to 1. The PORF flag is initialized by the RES pin reset.

Figure 5.1 shows examples of operations during a power-on reset and voltage monitor 0 reset.

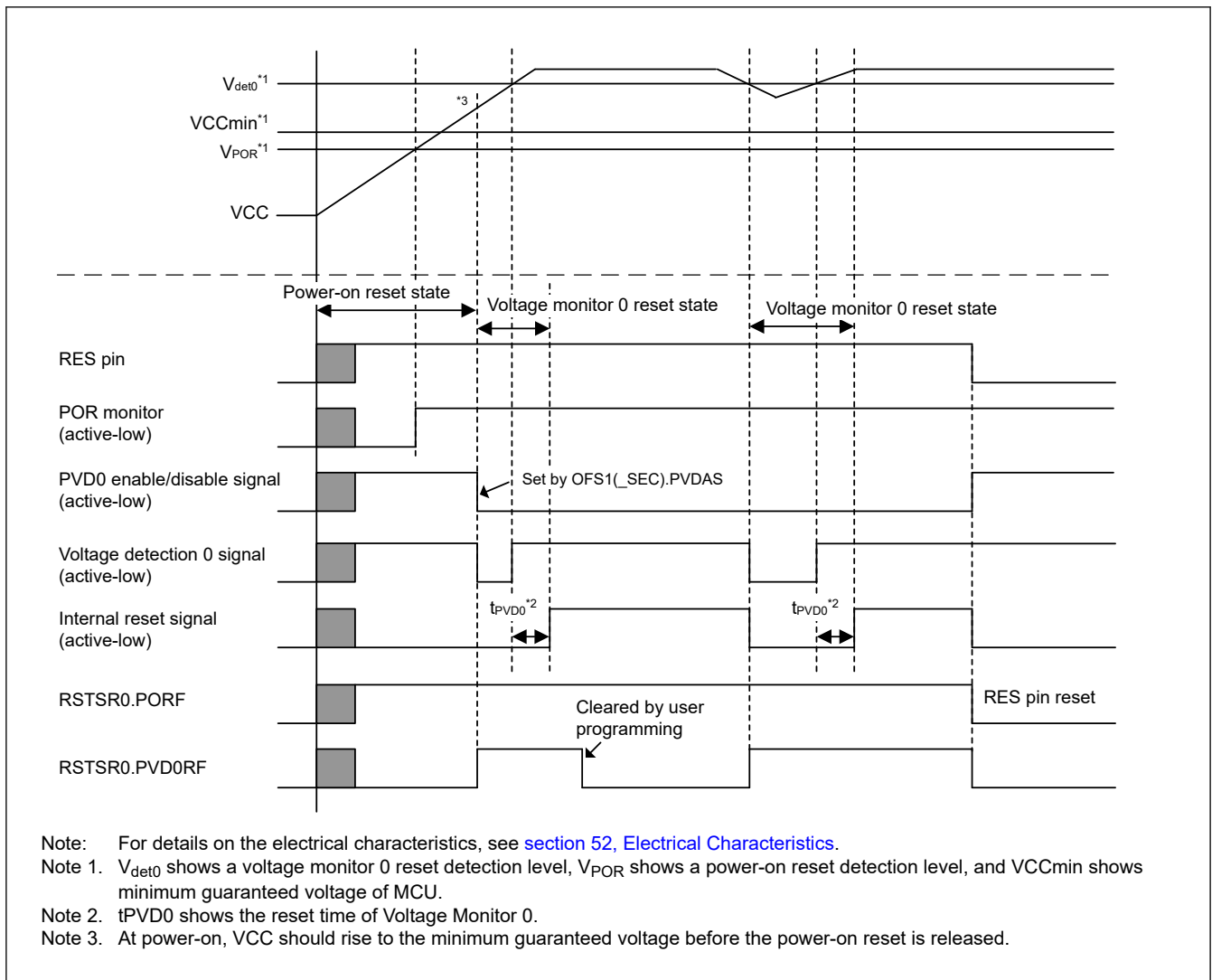


Figure 5.1 Example of operations during a power-on reset and Voltage Monitor 0 Reset

### 5.3.3 Voltage Monitor Reset

The voltage monitor 0 reset is an internal reset generated by the voltage monitor circuit. If the Voltage Detection 0 Circuit Start bit PVDAS) in Option Function Select Register 1 (OFS1) is 0 (voltage monitor 0 reset is enabled after a reset) and  $V_{CC}$  falls below  $V_{det0}$ , the RSTSR0.PVD0RF flag is set to 1 and the voltage detection circuit generates a voltage monitor 0 reset. Clear the OFS1(\_SEC).PVDAS bit to 0 if the voltage monitor 0 reset is to be used. After  $V_{CC}$  exceeds  $V_{det0}$  and the voltage monitor 0 reset time ( $t_{PVD0}$ ) elapses, the internal reset is canceled, and the CPU starts the reset exception handling. The  $V_{det0}$  voltage detection level can be changed by the setting in the VDSEL[1:0] bits in Option Function Select Register 1 (OFS1).

- Case of PVDmFCR.RHSEL = 0

When the Voltage Monitor 1 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 1 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 1 Circuit Control Register 0 (PVD1CR0), the RSTSR0.PVD1RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 1 reset if  $V_{CC}$  falls to or below  $V_{det1}$ .



Similarly, timing for release from the Voltage Monitor 1 reset state is selectable in the Voltage Monitor 1 Reset Negate Select bit (RN) in PVD1CR0. When the RN bit is 0 and VCC falls to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling when the PVD1 reset time ( $t_{PVD1}$ ) elapses after VCC rises above Vdet1. When the PVD1CR0.RN bit is 1 and VCC falls to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling when the PVD1 reset time ( $t_{PVD1}$ ) elapses.

Likewise, when the Voltage Monitor 2 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 2 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 2 Circuit Control Register 0 (PVD2CR0), the RSTSR0.PVD2RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 2 reset if VCC falls to or below Vdet2.

Similarly, timing for release from the Voltage Monitor 2 reset state is selectable in the Voltage Monitor 2 Reset Negate Select bit (RN) in PVD2CR0. When the RN bit is 0 and VCC falls to or below Vdet2, the CPU is released from the internal reset state and starts reset exception handling when the PVD2 reset time ( $t_{PVD2}$ ) elapses after VCC rises above Vdet2. When the PVD2CR0.RN bit is 1 and VCC falls to or below Vdet2, the CPU is released from the internal reset state and starts reset exception handling when the PVD2 reset time ( $t_{PVD2}$ ) elapses.

- Case of PVDmFCR.RHSEL = 1.

When the Voltage Monitor 1 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 1 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 1 Circuit Control Register 0 (PVD1CR0), the RSTSR0.PVD1RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 1 reset if VCC rises above Vdet1.

Then the CPU is released from the internal reset state and starts reset exception handling when the PVD1 reset time ( $t_{PVD1}$ ) elapses after VCC falls to or below Vdet1.

Likewise, when the Voltage Monitor 2 Interrupt/Reset Enable bit (RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the Voltage Monitor 2 Circuit Mode Select bit (RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in Voltage Monitor 2 Circuit Control Register 0 (PVD2CR0), the RSTSR0.PVD2RF flag is set to 1 and the voltage detection circuit generates a Voltage Monitor 2 reset if VCC rises above Vdet2.

Then the CPU is released from the internal reset state and starts reset exception handling when the PVD2 reset time ( $t_{PVD2}$ ) elapses after VCC falls to or below Vdet2.

Figure 5.2 shows examples of operations during the voltage monitor 1 and 2 reset when PVDmFCR.RHSEL=0.

Figure 5.3 shows examples of operations during the voltage monitor 1 and 2 reset when PVDmFCR.RHSEL=1.

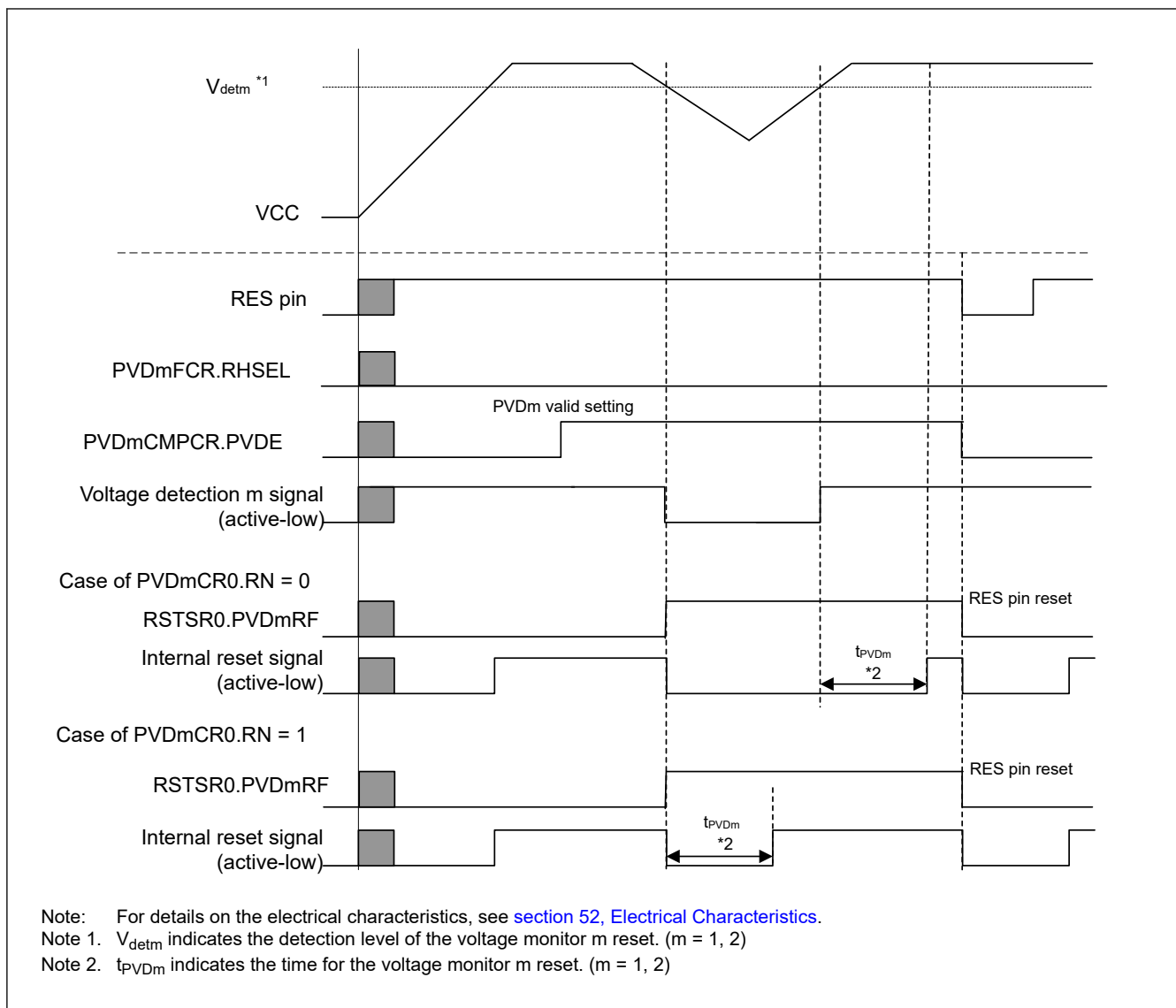


Figure 5.2 Example of operations during voltage monitor m (m = 1, 2) reset when PVDmFCR.RHSEL = 0

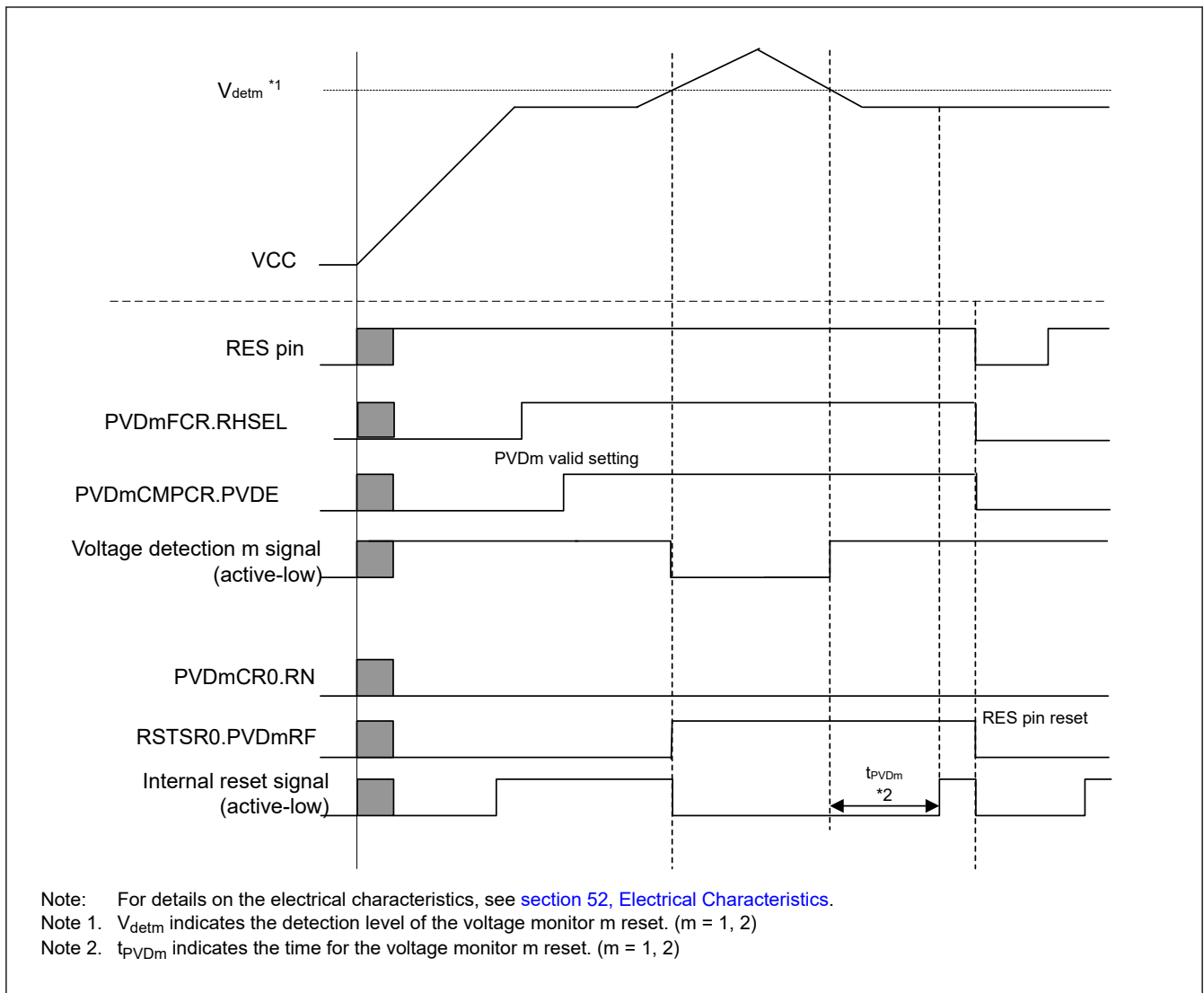


Figure 5.3 Example of operations during voltage monitor m (m = 1, 2) reset when PVDmFCR.RHSEL = 1

### 5.3.4 Deep Software Standby Reset

These internal resets are generated when either of the Deep Software Standby mode are canceled by an associated interrupt. The Deep Software Standby reset is canceled after  $t_{DSBY}$  (return time after Deep Software Standby mode cancellation) elapses. At the same time, Deep Software Standby mode is also canceled.

When  $t_{DSBYWT}$  (wait time after Deep Software Standby mode cancellation) elapses after Deep Software Standby mode has been canceled, the internal reset is canceled, and the CPU starts the reset exception handling.

For details of the Deep Software Standby reset, see [section 10, Low Power Modes](#).

### 5.3.5 Independent Watchdog Timer Reset

The independent watchdog timer reset is an internal reset generated from the Independent Watchdog Timer (IWDT). Output of the reset from the IWDT can be selected in the Option Function Select Register 0 (OFS0).

When output of the independent watchdog timer reset is selected, the reset is generated if the IWDT underflows, or if data is written when refresh operation is disabled. When the internal reset time ( $t_{RESW2}$ ) elapses after the independent watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the independent watchdog timer reset, see [section 26, Independent Watchdog Timer \(IWDT\)](#).

### 5.3.6 Watchdog Timer Reset

The watchdog timer reset is an internal reset generated from the Watchdog Timer (WDT). Output of the reset from the WDT can be selected in the WDT Reset Control Register (WDTRCR) or Option Function Select register 0 (OFS0).

When output of the watchdog timer reset is selected, the reset is generated if the WDT underflows, or if data is written when refresh operation is disabled. When the internal reset time ( $t_{RESW2}$ ) elapses after the watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the watchdog timer reset, see [section 25, Watchdog Timer \(WDT\)](#).

### 5.3.7 CPU Lockup Reset

The CPU Lockup reset is an internal reset generated by Arm core. Output of the reset from Arm core can be selected by CPULCKUPCR.OAD.

When output of the CPU Lockup reset is selected, CPU generated if Arm core is in the lockup state. When the internal reset time ( $t_{RESW2}$ ) elapses after the CPU Lockup reset is generated, the internal reset is cancelled, and the CPU starts the reset exception handling.

For details on the CPU Lockup reset, see [section 2, CPU](#).

### 5.3.8 Software Reset

This internal reset is generated by a software setting of the SYSRESETREQ bit in the AIRCR register in the Arm core. When the SYSRESETREQ bit is set to 1, a software reset is generated. When the internal reset time ( $t_{RESW2}$ ) elapses after the software reset is generated, the internal reset is canceled, and the CPU starts the reset exception handling.

### 5.3.9 Bus Error Reset

Bus error reset is an internal reset generated by buses.

It is an integrated reset generated by buses, and it consists of the following:

- Bus error reset (MSAU error, MMPU error, Illegal address error, Slave TrustZone Filter error, Slave Bus error, Bufferable write error)

Output of Bus error reset can be selected by OADCFG.OAD.

When the internal reset time ( $t_{RESW2}$ ) elapses after the Bus error reset is generated, the internal reset is cancelled, and the CPU starts the reset exception handling.

For details on the Bus error reset, see [section 14, Buses](#). Determination of reset generation source in Bus error reset is similar to determination of interrupt generation source, see [section 13, Interrupt Controller Unit \(ICU\)](#).

### 5.3.10 Common Memory Error Reset

Common memory error reset is an internal reset generated by SRAMs.

It is an integrated reset generated by SRAMs and consists of the following:

- SRAM error reset (Parity error)
- Standby SRAM error reset (Parity error)

Output of SRAM error reset can be selected by SRAMCR1.OAD. Output of Standby SRAM error reset can be selected by STBRAMCR.OAD.

When the internal reset time ( $t_{RESW2}$ ) elapses after the Common memory error reset is generated, the internal reset is cancelled, and the CPU starts the reset exception handling.

For details on the Common memory error reset, see [section 14, Buses](#). Determination of reset generation source in Common memory error reset is similar to determination of interrupt generation source, see [section 13, Interrupt Controller Unit \(ICU\)](#).

### 5.3.11 Determination of Cold/Warm Start

Read the CWSF flag in RSTSR2 to determine the cause of reset processing. This flag indicates whether a power-on reset caused the reset processing (cold start) or a reset signal input during operation caused the reset processing (warm start).

The CWSF flag is set to 0 when a power-on reset occurs (cold start), otherwise the flag is not set to 0. The flag is set to 1 when 1 is written to it through software. It is not set to 0 even on writing 0 to it.

Figure 5.4 shows an example of cold/warm start determination operation.

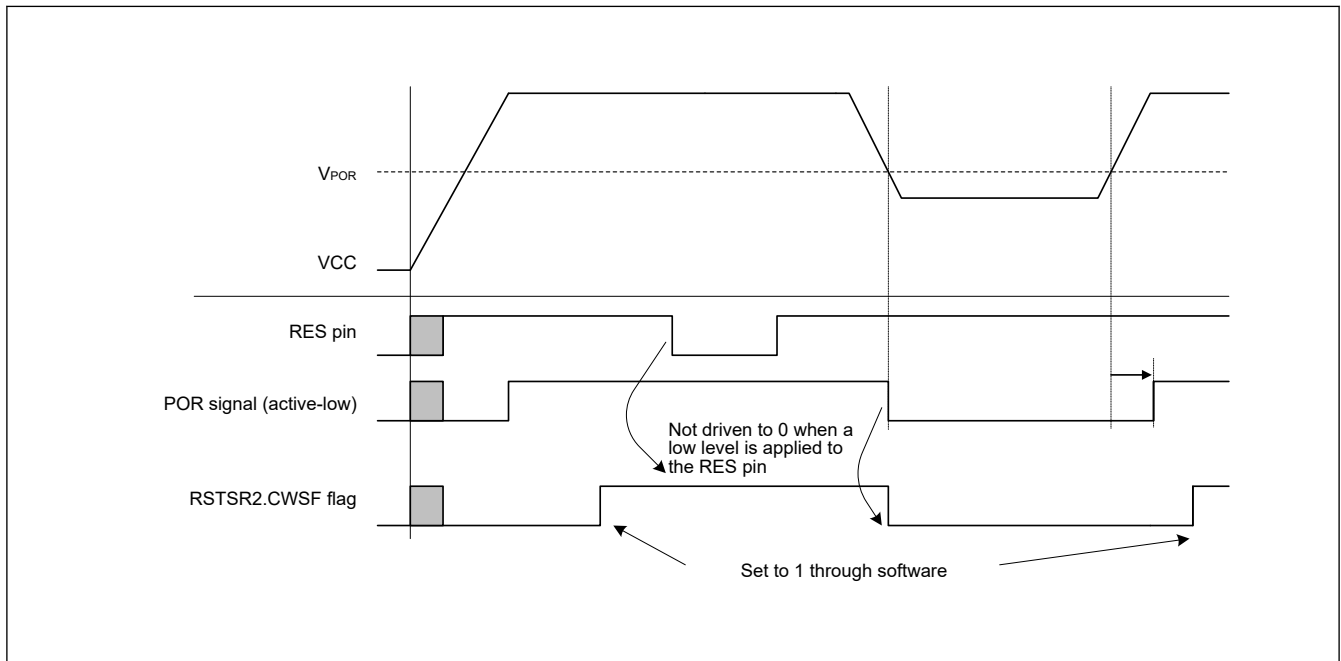


Figure 5.4 Example of cold/warm start determination operation

### 5.3.12 Determination of Reset Generation Source

Read RSTSR0 and RSTSR1 to determine which reset executes the reset exception handling.

Figure 5.5 shows an example of the flow to identify a reset generation source. The reset flag must be written with 0 after 1 is read.

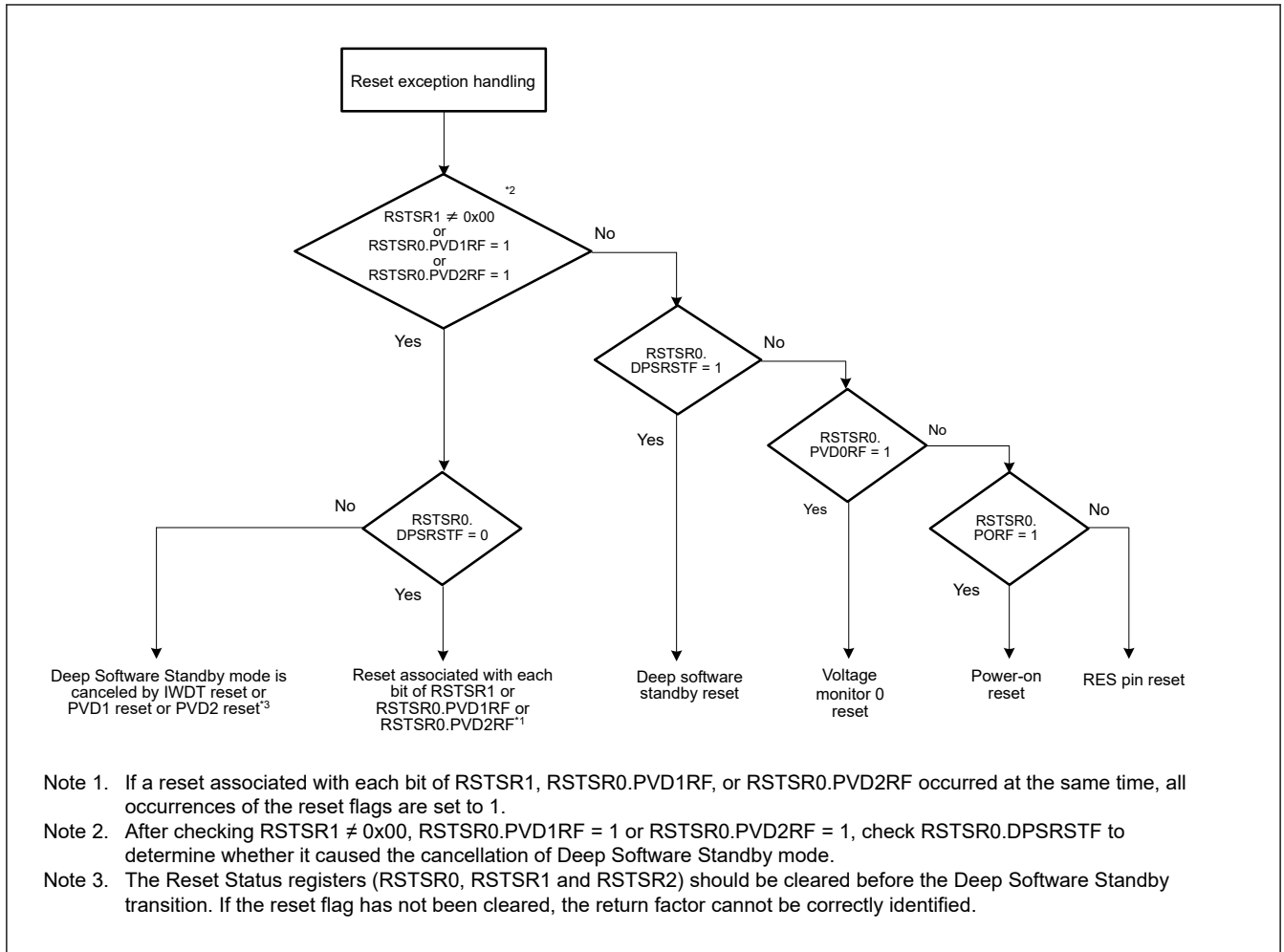


Figure 5.5 Example of reset generation source determination flow

### 5.4 Reset Protection from Non-secure Programs

The reset occurrence can be controlled. A reset is generated according to the setting of SYRSTMSK0 and SYRSTMSK2 that can only be accessed from a Secure program.

The reset source is generated but is not propagated to the MCU. The reset flags of RSTSR0 and RSTSR1 bits are not set.

The resets that can be controlled are as follows:

- Independent watchdog timer reset
- Watchdog timer reset
- CPU Lockup reset
- Common memory error reset
- Bus error reset
- Voltage monitor 1 reset
- Voltage monitor 2 reset
- Software reset

## 6. Option-Setting Memory

### 6.1 Overview

The option-setting memory determines the state of the MCU after a reset. The option-setting memory is allocated to the configuration setting area of the flash memory.

Figure 6.1 shows the option-setting memory area. The option-setting memory area has Secure region and Non-secure region. Table 6.1 shows the programming condition of the option-setting memory area.

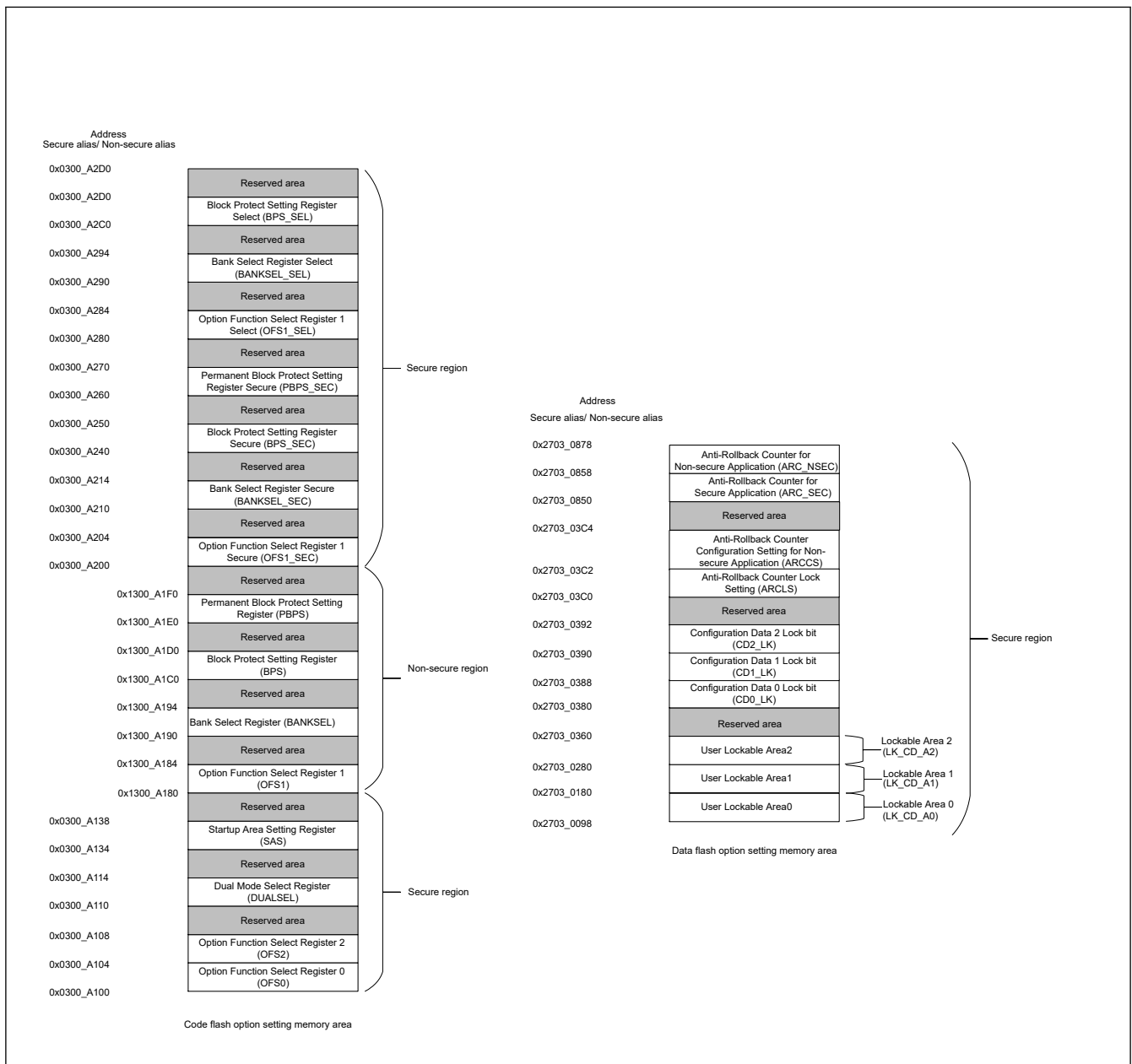


Figure 6.1 Option-setting memory area

Table 6.1 The programming condition of the option-setting memory area (1 of 2)

	Self programming	Serial programming	Programming by the on-chip debugger
Secure region	Programming commands issued via Secure alias access.	Programming commands issued when the authentication level is AL2.	Programming commands issued when the authentication level is AL2.

**Table 6.1 The programming condition of the option-setting memory area (2 of 2)**

	Self programming	Serial programming	Programming by the on-chip debugger
Non-secure region	Programming commands issued via Non-secure alias access.	Programming commands issued when the authentication level is AL2 or AL1.	Programming commands issued when the authentication level is AL2 or AL1.

## 6.2 Register Descriptions

### 6.2.1 OFS0 : Option Function Select Register 0

Address: 0x0300\_A100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	WDT0 STPC TL	—	WDT0 RSTIR QS	WDT0RPSS[1:0]	WDT0RPES[1:0]	WDT0CKS[3:0]			WDT0TOPS[1:0]	WDT0 STRT	—				

Value after reset: User setting\*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	IWDT STPC TL	—	IWDT RSTIR QS	IWDRPSS[1:0]	IWDRPES[1:0]	IWDTCKS[3:0]			IWDTTOPS[1:0]	IWDT STRT	—				

Value after reset: User setting\*1

Bit	Symbol	Function	R/W
0	—	The program value is read from this bit.*2	R
1	IWDTSTRT	IWDT Start Mode Select 0: Automatically activate IWDT after a reset (auto start mode) 1: Stop IWDT after a reset (register start mode)	R
3:2	IWDTTOPS[1:0]	IWDT Timeout Period Select 0 0: 128 cycles (0x007F) 0 1: 512 cycles (0x01FF) 1 0: 1024 cycles (0x03FF) 1 1: 2048 cycles (0x07FF)	R
7:4	IWDTCKS[3:0]	IWDT-Dedicated Clock Frequency Division Ratio Select 0x0: × 1 0x2: × 1/16 0x3: × 1/32 0x4: × 1/64 0xF: × 1/128 0x5: × 1/256 Others: Reserved	R
9:8	IWDRPES[1:0]	IWDT Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting)	R
11:10	IWDRPSS[1:0]	IWDT Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting)	R
12	IWDRSTIRQS	IWDT Reset Interrupt Request Select 0: Interrupt 1: Reset	R
13	—	The program value is read from this bit.*2	R



Bit	Symbol	Function	R/W
14	IWDTSTPCTL	IWDT Stop Control 0: Continue counting 1: Stop counting when in CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode or Deep Software Standby mode1	R
16:15	—	The program value is read from these bits.*2	R
17	WDT0STRT	WDT Start Mode Select 0: Automatically activate WDT after a reset (auto start mode) 1: Stop WDT after a reset (register start mode)	R
19:18	WDT0TOPS[1:0]	WDT Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R
23:20	WDT0CKS[3:0]	WDT Clock Frequency Division Ratio Select 0x1: PCLKB divided by 4 0x4: PCLKB divided by 64 0xF: PCLKB divided by 128 0x6: PCLKB divided by 512 0x7: PCLKB divided by 2048 0x8: PCLKB divided by 8192 Others: Reserved	R
25:24	WDT0RPES[1:0]	WDT Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (no window end position setting)	R
27:26	WDT0RPSS[1:0]	WDT Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (no window start position setting)	R
28	WDT0RSTIRQS	WDT Reset Interrupt Request Select 0: Interrupt 1: Reset	R
29	—	The program value is read from this bit.*2	R
30	WDT0STPCTL	WDT Stop Control 0: Continue counting 1: Stop counting when entering CPU Sleep mode or CPU Deep Sleep mode	R
31	—	The program value is read from this bit.*2	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

Note 2. This register can only be programmed by FACL command. When programming, the set value should be 1.

### IWDTSTRT bit (IWDT Start Mode Select)

The IWDTSTRT bit selects the mode in which the IWDT is activated after a reset (stopped state or activated state). When IWDT is activated in auto start mode, the OFS0 register setting for the IWDT is valid.

### IWDTTOPS[1:0] bits (IWDT Timeout Period Select)

The IWDTTOPS[1:0] bits specify the timeout period, that is, the time it takes for the down counter to underflow, as 128, 512, 1024, or 2048 cycles of the frequency-divided clock set in the IWDTCKS[3:0] bits. The time it takes for the counter to underflow after a refresh operation is determined by the combination of the IWDTCKS[3:0] and IWDTTOPS[1:0] bits.

For details, see [section 26, Independent Watchdog Timer \(IWDT\)](#).

### IWDTCKS[3:0] bits (IWDT-Dedicated Clock Frequency Division Ratio Select)

The IWDTCKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of the clock for the IWDT as 1/1, 1/16, 1/32, 1/64, 1/128, and 1/256. Using this setting combined with the IWDTTOPS[1:0] bits setting, the IWDT counting period can be set from 128 to 524,288 IWDT clock cycles.

For details, see [section 26, Independent Watchdog Timer \(IWDT\)](#).

#### **IWDRPES[1:0] bits (IWDT Window End Position Select)**

The IWDRPES[1:0] bits specify the position where the window for the down counter ends as 0%, 25%, 50%, or 75% of the count value. The value of the window end position must be smaller than the value of the window start position, otherwise only the value for the window start position is valid.

The counter values associated with the settings for the start and end positions of the window in the IWDRPSS[1:0] and IWDRPES[1:0] bits vary with the setting in the IWDTTOPS[1:0] bits.

For details, see [section 26, Independent Watchdog Timer \(IWDT\)](#).

#### **IWDRPSS[1:0] bits (IWDT Window Start Position Select)**

The IWDRPSS[1:0] bits specify the position where the window for the down counter starts as 25%, 50%, 75%, or 100% of the counted value. The point at which counting starts is 100% and the point at which an underflow occurs is 0%. The interval between the window starts and ends positions becomes the period in which a refresh is possible. Refresh is not possible outside this period.

For details, see [section 26, Independent Watchdog Timer \(IWDT\)](#).

#### **IWDRSTIRQS bit (IWDT Reset Interrupt Request Select)**

The IWDRSTIRQS bit selects the operation on an underflow of the down counter or generation of a refresh error. The operation is selectable to an independent watchdog timer reset, a non-maskable interrupt request, or an interrupt request.

For details, see [section 26, Independent Watchdog Timer \(IWDT\)](#).

#### **IWDTSTPCTL bit (IWDT Stop Control)**

The IWDTSTPCTL bit specifies whether to stop counting when entering CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode or Deep Software Standby mode1.

For details, see [section 26, Independent Watchdog Timer \(IWDT\)](#).

#### **WDT0STRT bit (WDT Start Mode Select)**

The WDT0STRT bit selects the mode in which the WDT is activated after a reset (stopped state or activated in auto start mode). When WDT is activated in auto start mode, the OFS0 register setting for the WDT is valid.

#### **WDT0TOPS[1:0] bits (WDT Timeout Period Select)**

The WDT0TOPS[1:0] bits specify the timeout period, that is, the time it takes for the down counter to underflow as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set in the WDT0CKS[3:0] bits. The number of PCLKB cycles that takes to underflow after a refresh operation is determined by a combination of the WDT0CKS[3:0] and WDT0TOPS[1:0] bits.

For details, see [section 25, Watchdog Timer \(WDT\)](#).

#### **WDT0CKS[3:0] bits (WDT Clock Frequency Division Ratio Select)**

The WDT0CKS[3:0] bits specify the division ratio of the prescaler for dividing the frequency of PCLKB as 1/4, 1/64, 1/128, 1/512, 1/2048, and 1/8192. Using this setting combined with the WDT0TOPS[1:0] bits setting, the WDT counting period can be set from 4,096 to 134,217,728 PCLKB cycles.

For details, see [section 25, Watchdog Timer \(WDT\)](#).

#### **WDT0RPES[1:0] bits (WDT Window End Position Select)**

The WDT0RPES[1:0] bits specify the position where the window on the down counter ends as 0%, 25%, 50%, or 75% of the counted value. The value of the window end position must be smaller than the value of the window start position, otherwise only the value for the window start position is valid.

The counter values associated with the settings for the start and end positions of the window in the WDT0RPSS[1:0] and WDT0RPES[1:0] bits vary with the setting of the WDT0TOPS[1:0] bits.

For details, see [section 25, Watchdog Timer \(WDT\)](#).

**WDT0RPSS[1:0] bits (WDT Window Start Position Select)**

The WDT0RPSS[1:0] bits specify the position where the window for the down counter starts as 25%, 50%, 75%, or 100% of the counted value. The point at which counting starts is 100% and the point at which an underflow occurs is 0%. The interval between the positions where the window starts and ends becomes the period in which a refresh is possible.

Refresh is not possible outside this period.

For details, see [section 25, Watchdog Timer \(WDT\)](#).

**WDT0RSTIRQS bit (WDT Reset Interrupt Request Select)**

The WDT0RSTIRQS bit selects the operation on an underflow of the down-counter or generation of a refresh error. The operation is selectable to a watchdog timer reset, a non-maskable interrupt request, or an interrupt request.

For details, see [section 25, Watchdog Timer \(WDT\)](#).

**WDT0STPCTL bit (WDT Stop Control)**

The WDT0STPCTL bit specifies whether to stop counting when entering CPU Sleep mode or CPU Deep Sleep mode.

For details, see [section 25, Watchdog Timer \(WDT\)](#).

**6.2.2 OFS2 : Option Function Select Register 2**

address: 0x0300\_A104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Value after reset: User setting\*<sup>1</sup>

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DCDC EN

Value after reset: User setting\*<sup>1</sup>

Bit	Symbol	Function	R/W
0	DCDCEN	DCDC enable 0: Disable DCDC 1: Enable DCDC	R
31:1	—	The program value is read from these bits.* <sup>2</sup>	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

**DCDCEN bit (DCDC enable)**

The DCDCEN bit selects whether the DCDC is enabled or disabled.

**6.2.3 DUALSEL : Dual Mode Select Register**

address: 0x0300\_A110

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Value after reset: User setting\*<sup>1</sup>

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BANKMD[2:0]		

Value after reset: User setting\*<sup>1</sup>

Bit	Symbol	Function	R/W
2:0	BANKMD[2:0]	Bank Mode Select 0 0 0: Dual mode 1 1 1: Linear mode Others: Reserved	R
31:3	—	The program value is read from these bits. <sup>*2</sup>	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application

Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

### BANKMD[2:0] bit (Bank Mode Select)

The BANKMD[2:0] bits select bank mode of the dual bank function of the code flash memory

## 6.2.4 SAS : Startup Area Setting Register

Address: 0x0300\_A134

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BTFLG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	User setting <sup>*1</sup>															
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FSPR	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	User setting <sup>*1</sup>															

Bit	Symbol	Function	R/W
14:0	—	The program value is read from these bits. <sup>*2</sup>	R
15	FSPR	Protection of Startup Area Select Function This bit controls the programming of the write/erase protection for the Startup Area Select flag (SAS.BTFLG), and the temporary boot swap control. When this bit is set to 0, it cannot be changed to 1. 0: Executing the configuration setting command for programming the Startup Area Select flag (SAS.BTFLG) is invalid. 1: Executing the configuration setting command for programming the Startup Area Select flag (SAS.BTFLG) is valid.	R
30:16	—	The program value is read from these bits. <sup>*2</sup>	R
31	BTFLG	Startup Area Select Flag This bit specifies whether the address of the startup area is exchanged for the boot swap function or not. In dual mode (the DUALSEL.BANKMD[2:0] bits are 000b), set 1 to this bit. 0: First 8 KB area (0x0200_0000 to 0x0200_1FFF) and second 8 KB area (0x0200_2000 to 0x0200_3FFF) are exchanged. 1: First 8 KB area (0x0200_0000 to 0x0200_1FFF) and second 8 KB area (0x0200_2000 to 0x0200_3FFF) are not exchanged.	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application

Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

### 6.2.5 OFS1, OFS1\_SEC : Option Function Select Register 1 for Non-secure and Secure

Address: OFS1: 0x1300\_A180 (Non-secure)  
 OFS1\_SEC: 0x0300\_A200 (Secure)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	INITECCEN	SWDBG	—	—	—	—	—	—	—	—

Value after reset: The value set by the user\*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	HOCOFREQ[2:0]	HOCOEN	—	—	PVDLPSEL	—	PVDAS	—	—	—	—	—

Value after reset: The value set by the user\*1

Bit	Symbol	Function	R/W
2:0	VDSEL[2:0]	Voltage Detection 0 Level Select 0 0 0: Select 2.85 V 0 0 1: Select 2.58 V 0 1 0: Select 2.15 V 0 1 1: Select 2.00 V 1 0 0: Select 1.90 V 1 0 1: Select 1.80 V 1 1 0: Select 1.70 V 1 1 1: Select 1.60 V*3	R
3	PVDAS	Voltage Detection 0 Circuit Start 0: Enable voltage monitor 0 reset after a reset 1: Disable voltage monitor 0 reset after a reset	R
4	—	The program value is read from this bit.*2	R
5	PVDLPSEL	Low Power Consumption Function of PVD0 Select at Deep Software Standby mode 0: Enable low power consumption function of PVD0 during DSTBY1 and DSTBY2. 1: Disable low power consumption function of PVD0 during DSTBY1 and DSTBY2.	R
7:6	—	The program value is read from these bits.*2	R
8	HOCOEN	HOCO Oscillation Enable 0: Enable HOCO oscillation after a reset 1: Disable HOCO oscillation after a reset	R
11:9	HOCOFREQ[2:0]	HOCO Frequency Setting 0 0 0 0: 16 MHz 0 0 1: 18 MHz 0 1 0: 20 MHz 1 0 0: 32 MHz 1 1 1: 48 MHz Others: Reserved.	R
23:12	—	The program value is read from these bits.*2	R
24	SWDBG	Software Debug Control 0: Enable software debug control. The MCU behaves according to DBGAUTH0 and DBGAUTH1, and IWDT and WDT automatically stops when CPU is in the debug state. 1: Disable software debug control. The MCU ignores DBGAUTH0 and DBGAUTH1, and IWDT and WDT continue the operation even when CPU is in the debug state. However, when CDBGPWRUPREQ is 1 and AL is not AL0, IWDT and WDT automatically stops when CPU is in the debug state.	R
25	INITECCEN	Initial ECC Enable 0: Disable ECC function of TCM and CACHE. 1: Enable ECC function of TCM and CACHE.	R
31:26	—	The program value is read from these bits.*2	R

Note 1. The value of OFS1 and OFS1\_SEC in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

Note 2. This register can only be programmed by FACL command. When programming, the set value should be 1.

Note 3. Setting prohibited when VBATT function is enabled, OFS1(\_SEC).PVDAS and PVDLPSEL bits are 0 and at least one of Deep Software Standby mode 1 and 2 is used.

OFS1 register is for Non-secure developer, and OFS1\_SEC register is for Secure developer.

### VDSEL[2:0] bits (Voltage Detection 0 Level Select)

The VDSEL[2:0] bits select the voltage detection level of the voltage detection 0 circuit.

### PVDAS bits (Voltage Detection 0 Circuit Start)

The PVDAS bit selects whether the voltage monitor 0 reset is enabled or disabled after a reset.

### PVDLPSEL bits (Low Power Consumption Function of PVD0 Select at Deep Software Standby mode)

The PVDLPSEL selects whether the low power consumption function of PVD0 is enabled or disabled during DSTBY1 and DSTBY2 mode.

When disable this function, the response time is faster instead of increasing current consumption. When enable this function, the current consumption is smaller instead of response time delay.

See [section 52, Electrical Characteristics](#)

### HOCOEN bit (HOCO Oscillation Enable)

The HOCOEN bit selects whether the HOCO oscillation is enabled or disabled after a reset. Setting this bit to 0 allows the HOCO oscillation to start before the CPU starts operation, which reduces the wait time for oscillation stabilization.

Note: When the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is only switched to HOCO by setting the Clock Source Select bits (SCKSCR.CKSEL[2:0]). To use the HOCO clock, Set the OFS1(\_SEC).HOCOFRQ[2:0] bits\*<sup>1</sup> to an optimum value.

Note 1. OFS1 is for non-secure developers and OFS1\_SEC is for secure developers. The applied setting value is determined by OFS1\_SEL. The value of OFS1.HOCOFRQ[2:0] bits is automatically transferred to HOCOCR2.HCFRQ[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ[2:0] bits when OFS1(\_SEC).HOCOEN=1.

### HOCOFRQ[2:0] bits (HOCO Frequency Setting 0)

The HOCOFRQ[2:0] bits specify the HOCO frequency after a reset as 16 to 48 MHz.

### SWDBG bit (Software Debug Control)

The SWDBG bit selects whether software debug control is enabled or disabled.

### INITECCEN bit (Initial ECC Enable)

The INITECCEN bit selects whether ECC function of TCM and CACHE is enabled or disabled.

When the INITECCEN bit is changed from 1 to 0, be sure to perform a power-on reset after changing it.

## 6.2.6 OFS1\_SEL : Option Function Select Register 1 for Security Attribution

Address: OFS1\_SEL: 0x0300\_A280

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	INITECCEN	SWDBG	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user* <sup>1</sup>															
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	HOCOFRQ[2:0]			HOCOEN	—	—	PVDLPSEL	—	PVDAS	VDSEL[2:0]		
Value after reset:	The value set by the user* <sup>1</sup>															

Bit	Symbol	Function	R/W
2:0	VDSEL[2:0]	Security attributes of Voltage Detection 0 Level Select 0 0 0: Select OFS1_SEC.VDSEL[2:0] 1 1 1: Select OFS1.VDSEL[2:0] Others: Select Reserved.	R
3	PVDAS	Security attributes of Voltage Detection 0 Circuit Start 0: Select OFS1_SEC.PVDAS 1: Select OFS1.PVDAS	R
4	—	The program value is read from this bit.	R
5	PVDLPSEL	Security attributes of Low Power Consumption Function of PVD0 Select at Deep Software standby mode 0: Select OFS1_SEC.PVDLPSEL 1: Select OFS1.PVDLPSEL	R
7:6	—	The program value is read from these bits.	R
8	HOCOEN	Security attributes of HOCO Oscillation Enable 0: Select OFS1_SEC.HOCOEN 1: Select OFS1.HOCOEN	R
11:9	HOCOFRQ0[2:0]	Security attributes of HOCO Frequency Setting 0 0 0 0: Select OFS1_SEC.HOCOFRQ0[2:0] 1 1 1: Select OFS1.HOCOFRQ0[2:0] Others: Reserved	R
23:12	—	The program value is read from these bits.	R
24	SWDBG	Security attributes of Software Debug Control 0: Select OFS1_SEC.SWDBG 1: Select OFS1.SWDBG	R
25	INITECCEN	Security attributes of Initial ECC Enable 0: Select OFS1_SEC.INITECCEN 1: Select OFS1.INITECCEN	R
31:26	—	The program value is read from these bits.	R

Note 1. The value of OFS1\_SEL in a blank product is 0x00000000. It is set to the value written by your application

The OFS1\_SEL register selects the security attribute. Which security attribute is applied, OFS or OFS\_SEC, is determined by the setting value of the corresponding bit in the OFS1\_SEL register. For details, see [section 6.3.3. Security attribution of code flash option-setting memory](#).

### 6.2.7 BANKSEL, BANKSEL\_SEC : Bank Select Register for Non-secure and Secure

Address: BANKSEL: 0x1300\_A190 (Non-secure)  
BANKSEL\_SEC: 0x0300\_A210 (Secure)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	BLCKSWP[3:0]			

Value after reset: User setting\*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BANKSWP[2:0]		

Value after reset: User setting\*1

Bit	Symbol	Function	R/W
2:0	BANKSWP[2:0]	Startup Bank Switch This setting is valid in dual mode. 0 0 0: Start address of Bank0 is code flash base address + 0x0020_0000 and Bank is code flash base address + 0x0000_0000 in dual mode 1 1 1: Start address of Bank0 is code flash base address + 0x0000_0000 and Bank is code flash base address + 0x0020_0000 in dual mode Others: Reserved	R
15:3	—	The program value is read from these bits. *1	R
19:16	BLCKSWP[3:0]	Block Swap Select When all bits are set to 1, the block swap is disabled. When at least one bit is set to 0, block swap is enabled and the corresponding blocks of code flash memory are swapped.	R
31:20	—	The program value is read from these bits.*1	R

Note 1. The value of BANKSEL and BANKSEL\_SEC in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

BANKSEL register is for Non-secure developer, and BANKSEL\_SEC register is for Secure developer.

### BANKSWP[2:0] bits (Startup Bank Switch)

The BANKSWP[2:0] bits select the start address of the bank0 and bank1 of code flash memory in dual mode. For details of the startup bank selection, see [section 49, Flash Memory](#).

### BLCKSWP[3:0] bits (Block Swap Select)

The BLCKSWP[3:0] bits enable the block swap and select the valid blocks of the code flash memory in linear mode. [Figure 6.2](#) shows the mapping of the flash memory in linear mode. [Table 6.2](#) shows the specification of BLCKSWP bits for each product. Unused bits are reserved and should be set to 1. For details of block swap, see [section 49, Flash Memory](#).

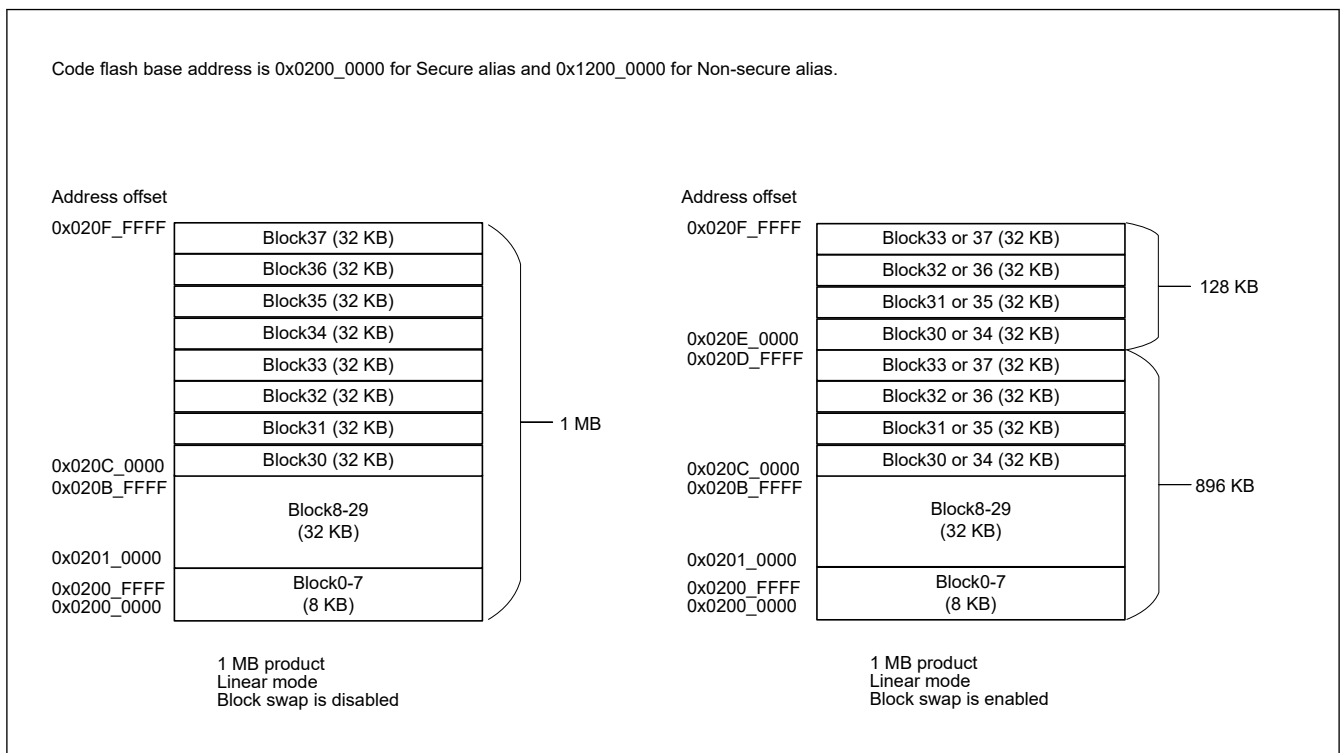


Figure 6.2 Mapping of the flash memory



**Table 6.2 Specification of BLCKSWP bits for each product**

	BLCKSWP bits	Select 0/1: Start address offset for each Block
1 MB product	BLCKSWP[0]	1 : Block30 start address is 0x000C_0000. Block34 start address is 0x000E_0000. 0 : Block30 start address is 0x000E_0000. Block34 start address is 0x000C_0000.
	BLCKSWP[1]	1 : Block31 start address is 0x000C_8000. Block35 start address is 0x000E_8000. 0 : Block31 start address is 0x000E_8000. Block35 start address is 0x000C_8000.
	BLCKSWP[2]	1 : Block32 start address is 0x000D_0000. Block36 start address is 0x000F_0000. 0 : Block32 start address is 0x000F_0000. Block36 start address is 0x000D_0000.
	BLCKSWP[3]	1 : Block33 start address is 0x000D_8000. Block37 start address is 0x000F_8000. 0 : Block33 start address is 0x000F_8000. Block37 start address is 0x000D_8000.

### 6.2.8 BANKSEL\_SEL : Bank Select Register for Security Attribution

Address: BANKSEL\_SEL: 0x0300\_A290

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	BLCKSWP[3:0]			

Value after reset: User setting\*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BANKSWP[2:0]		

Value after reset: User setting\*1

Bit	Symbol	Function	R/W
2:0	BANKSWP[2:0]	Security Attributes of Startup Bank Switch 0 0 0: Select BANKSEL_SEC.BANKSWP[2:0] 1 1 1: Select BANKSEL.BANKSWP[2:0] Others: Reserved	R
15:3	—	The program value is read from these bits. *1	R
19:16	BLCKSWP[3:0]	Security Attributes of Block Swap Select BANKSEL_SEL.BLCKSWP[3:0] bits can select Secure or Non-secure access for each bit. The bit positions of BLCKSWP[3:0] in the BANKSEL_SEL register and the bit positions of BLCKSWP[3:0] in the BANKSEL and BANKSEL_SEC registers correspond to indicate the same block. For example, when BANKSEL_SEL.BLCKSWP[0] = 0 and BANKSEL_SEL.BLCKSWP[3] = 1, then BANKSEL_SEC.BLCKSWP[0] and BANKSEL.BLCKSWP[3] are selected.	R
31:20	—	The program value is read from these bits.. *1	R

Note 1. The value of BANKSEL\_SEL in a blank product is 0x00000000. It is set to the value written by your application

Note 2. This register can only be programmed by FACI command. When programming, the set value should be 0

The BANKSEL\_SEL register selects the security attribute. Which security attribute is applied, BANKSEL or BANKSEL\_SEC, is determined by the setting value of corresponding bit in the BANKSEL\_SEL register. For details, see

[section 6.3.3. Security attribution of code flash option-setting memory](#)

### 6.2.9 BPS, BPS\_SEC, BPS\_SEL : Block Protect Setting Register

address:

BPS: 0x1300\_A1C0, 0x1300\_A1C4, 0x1300\_A1C8 (Non-secure)  
 BPS\_SEC: 0x0300\_A240, 0x0300\_A244, 0x0300\_A248 (Secure)  
 BPS\_SEL: 0x0300\_A2C0, 0x0300\_A2C4, 0x0300\_A2C8 (Secure)

Bit position: 31

0

Bit field:

Value after reset:

User setting\*1

Note 1. The value of BPS and BPS\_SEC in a blank product is 0xFFFFFFFF. The value of BPS\_SEL in factory shipment product is 0x00000000. It is set to the value written by your application.

BPS\_SEC and BPS\_SEL register is for Secure developer, and BPS register is for Non-secure developer.

The BPS and BPS\_SEC registers invalidate the programming and erasure to the code flash memory. When the bit of this register is set to 0, the programming and erasure to the corresponding block are invalid. [Figure 6.3](#) shows the code flash block structure of each product. [Figure 6.4](#) shows the relationship between the bit of register and the block number. Unused bits of BPS and BPS\_SEC register are reserved and should be programmed to 1.

The BPS\_SEL register selects the security attribute. Whether the security attribute BPS or BPS\_SEC, PBPS or PBPS\_SEC is applied is determined by the setting value of the corresponding bit in the BPS\_SEL register. For details, see [section 6.3.3. Security attribution of code flash option-setting memory](#).

Code flash base address is 0x0200\_0000 for Secure alias and 0x1200\_0000 for Non-secure alias.

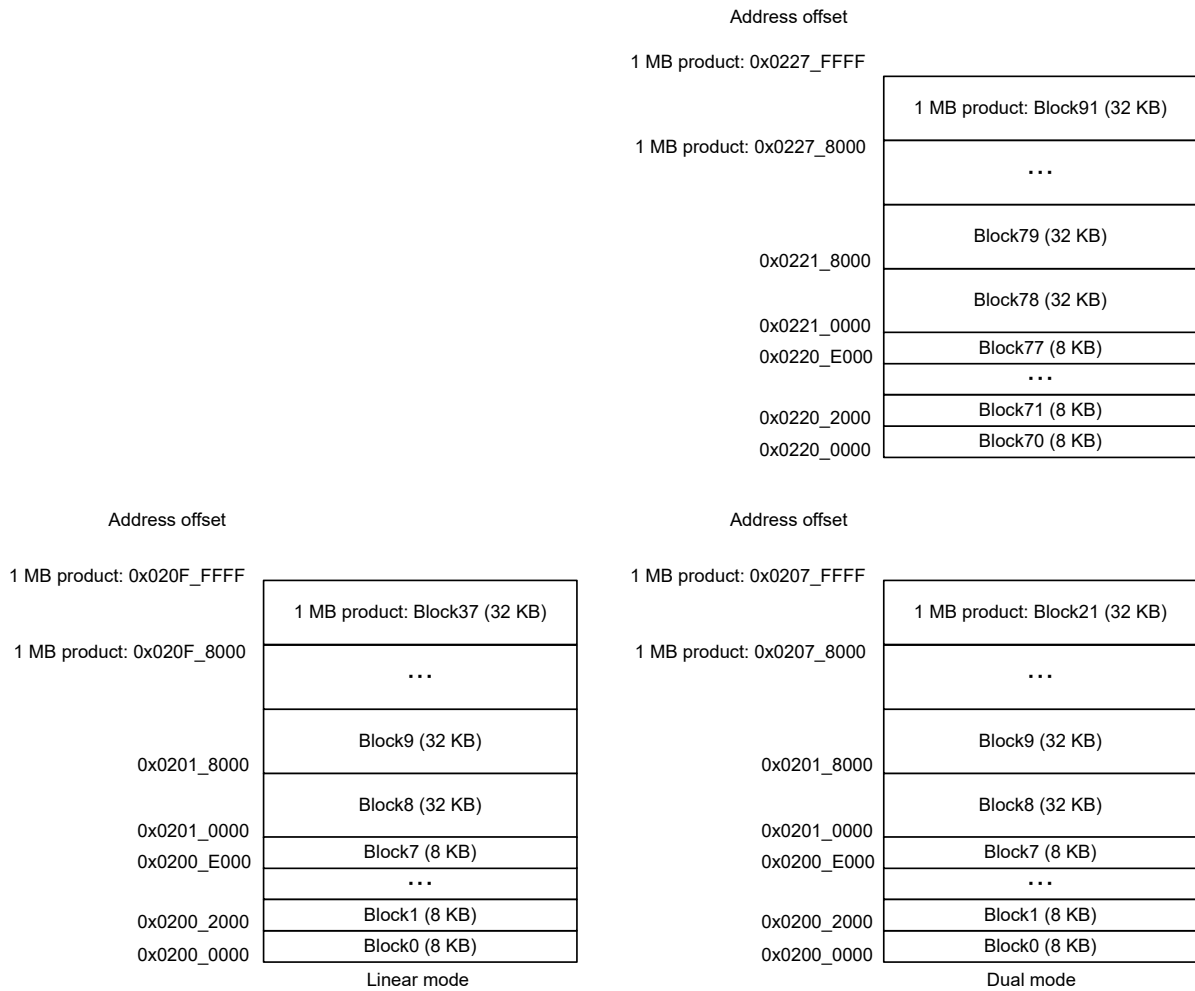


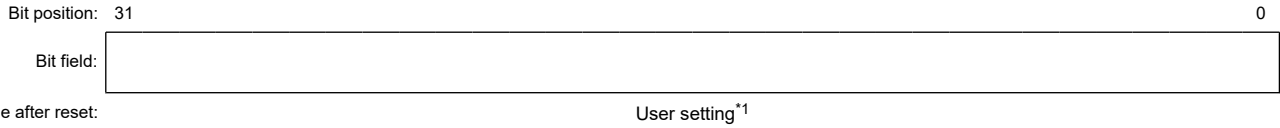
Figure 6.3 Code Flash block structure

Register	Address	+31	+30	+29	+28	+27	+26	+25	+24	+23	+22	+21	+20	+19	+18	+17	+16	+15	+14	+13	+12	+11	+10	+9	+8	+7	+6	+5	+4	+3	+2	+1	+0
BPS_SEL	0x0300_A2C8	-	-	-	-	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	-	-	-	-	-	-
	0x0300_A2C4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	37	36	35	34	33	32
	0x0300_A2C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I																																	
BPS_SEC	0x0300_A248	-	-	-	-	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	-	-	-	-	-	-
	0x0300_A244	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	37	36	35	34	33	32
	0x0300_A240	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I																																	
BPS	0x1300_A1C8	-	-	-	-	91	90	89	88	87	86	85	84	83	82	81	80	79	78	77	76	75	74	73	72	71	70	-	-	-	-	-	-
	0x1300_A1C4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	37	36	35	34	33	32
	0x1300_A1C0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 6.4 The relationship between the bit of register and the block number

### 6.2.10 PBPS, PBPS\_SEC : Permanent Block Protect Setting Register

Address: PBPS: 0x1300\_A1E0, 0x1300\_A1E4, 0x1300\_A1E8 (Non-secure)  
 PBPS\_SEC: 0x0300\_A260, 0x0300\_A264, 0x0300\_A268 (Secure)



Note 1. The value of PBPS and PBPS\_SEC in a blank product is 0xFFFFFFFF. It is set to the value written by your application.

PBPS\_SEC register is for secure developer, and PBPS register is for non-secure developer. The applied setting value is determined by the setting value of the corresponding bit in BPS\_SEL register. For details, see [section 6.3.3. Security attribution of code flash option-setting memory](#). The security attribution register is same BPS\_SEL register between the block protection and permanent block protection.

The PBPS and PBPS\_SEC registers invalidate writes to bits of BPS and BPS\_SEC. The bit of this register can be programmed 0 when corresponding bit of BPS and BPS\_SEC is programmed to 0. When the bit of this register is programmed to 0, writing the corresponding bit of BPS and BPS\_SEC register is invalid. Once the bit of this register is set to 0, it is impossible to change the bit to 1. [Table 6.3](#) shows the relationship between the bit of applied PBPS and bit of applied BPS.

When the applied setting value of this register selected by BPS\_SEL register is 0, corresponding bit of BPS\_SEL register cannot be changed.

The relationship between the bit of this register and the block number is same as BPS and BPS\_SEC registers ([Figure 6.4](#)). Unused bits are reserved and should be set to 1.

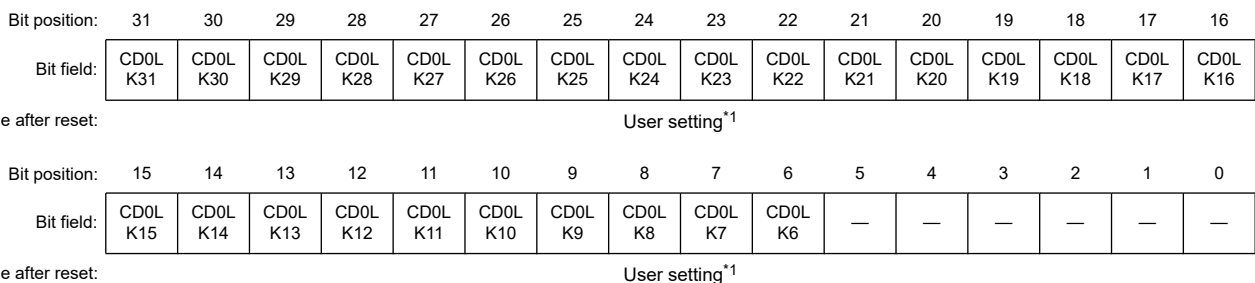
**Table 6.3 The relationship between the bit of PBPS, PBPS\_SEC and bit of BPS, BPS\_SEC**

The bit of applied PBPS and PBPS_SEC	The bit of applied BPS and BPS_SEC	Content
1	1	Programming and erasure to the corresponding block is valid.
1	0	Programming and erasure to the corresponding block is invalid. This protection can be canceled by FBPROT0 or FBPROT1 registers.
0	1	Can not set this condition
0	0	Programming and erasure to the corresponding block is invalid permanently

Note: When BPS\_SEL = 0, BPS and PBPS correspond. When BPS\_SEL = 1, BPS\_SEC and PBPS\_SEC correspond.

### 6.2.11 CFGD0LOCK : Configuration Data 0 Lock Bit

Base address: 0x2703\_0380



Base address: 0x2703\_0384

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CD0L K63	CD0L K62	CD0L K61	CD0L K60	CD0L K59	CD0L K58	CD0L K57	CD0L K56	CD0L K55	CD0L K54	CD0L K53	CD0L K52	CD0L K51	CD0L K50	CD0L K49	CD0L K48

Value after reset: User setting\*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CD0L K47	CD0L K46	CD0L K45	CD0L K44	CD0L K43	CD0L K42	CD0L K41	CD0L K40	CD0L K39	CD0L K38	CD0L K37	CD0L K36	CD0L K35	CD0L K34	CD0L K33	CD0L K32

Value after reset: User setting\*1

Bit	Symbol	Function	R/W
5:0	—	The program value is read from these bits.	R
31:6	CD0LK31 to CD0LK6	Configuration Data 0 Lock Bit 0: LK_CD_A0 protection is enabled 1: LK_CD_A0 protection is disabled	R
31:0	CD0LK63 to CD0LK32	Configuration Data 0 Lock Bit 0: LK_CD_A0 protection is enabled 1: LK_CD_A0 protection is disabled	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application

### CD0LKn bits (Configuration Data 0 Lock Bit)

These bits specify write protection for Lockable Configuration Data Area 0 (LK\_CD\_A0) in configuration area of data flash. CD0\_LK(n) protects 4 bytes data in LK\_CD\_A0.

CD0\_LK bits can be set to "0" by the "Configuration set" command and cannot be changed from "0" to "1".

### 6.2.12 CFGD1LOCK : Configuration Data 1 Lock Bit

Base address: 0x2703\_0388

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CD1L K31	CD1L K30	CD1L K29	CD1L K28	CD1L K27	CD1L K26	CD1L K25	CD1L K24	CD1L K23	CD1L K22	CD1L K21	CD1L K20	CD1L K19	CD1L K18	CD1L K17	CD1L K16

Value after reset: User setting\*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CD1L K15	CD1L K14	CD1L K13	CD1L K12	CD1L K11	CD1L K10	CD1L K9	CD1L K8	CD1L K7	CD1L K6	CD1L K5	CD1L K4	CD1L K3	CD1L K2	CD1L K1	CD1L K0

Value after reset: User setting\*1

Base address: 0x2703\_038C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CD1L K63	CD1L K62	CD1L K61	CD1L K60	CD1L K59	CD1L K58	CD1L K57	CD1L K56	CD1L K55	CD1L K54	CD1L K53	CD1L K52	CD1L K51	CD1L K50	CD1L K49	CD1L K48

Value after reset: User setting\*1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CD1L K47	CD1L K46	CD1L K45	CD1L K44	CD1L K43	CD1L K42	CD1L K41	CD1L K40	CD1L K39	CD1L K38	CD1L K37	CD1L K36	CD1L K35	CD1L K34	CD1L K33	CD1L K32

Value after reset: User setting\*1

Bit	Symbol	Function	R/W
31:0	CD1LK31 to CD1LK0	Configuration Data 1 Lock Bit 0: LK_CD_A1 protection is enabled 1: LK_CD_A1 protection is disabled	R
31:0	CD1LK63 to CD1LK32	Configuration Data 1 Lock Bit 0: LK_CD_A1 protection is enabled 1: LK_CD_A1 protection is disabled	R

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application

### CD1LK<sub>n</sub> bits (Configuration Data 1 Lock Bit)

These bits specify write protection for Lockable Configuration Data Area 1 (LK\_CD\_A1) in configuration area of data flash. CD1\_LK(n) protects 4 bytes data in LK\_CD\_A1.

CD1\_LK bits can be set to "0" by the "Configuration set" command and cannot be changed from "0" to "1"

### 6.2.13 CFGD2LOCK : Configuration Data 2 Lock Bit

Base address: 0x2703\_0390

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CD2L K13	CD2L K12	CD2L K11	CD2L K10	CD2L K9	CD2L K8	CD2L K7	CD2L K6	CD2L K5	CD2L K4	CD2L K3	CD2L K2	CD2L K1	CD2L K0

Value after reset: User setting\*1

Bit	Symbol	Function	R/W
13:0	CD2LK13 to CD2LK0	Configuration Data 2 Lock Bit 0: LK_CD_A2 protection is enabled 1: LK_CD_A2 protection is disabled	R
15:14	—	The program value is read from these bits.	R

Note 1. The value in a blank product is 0xFFFF\_FFFF. It is set to the value written by your application

### CD2LK<sub>n</sub> bits (Configuration Data 2 Lock Bit)

These bits specify write protection for Lockable Configuration Data Area 2 (LK\_CD\_A2) in configuration area of data flash. CD2\_LK(n) protects 16 bytes data in LK\_CD\_A2. CD2\_LK bits can be set to "0" by the "Configuration set" command and cannot be changed from "0" to "1".

Settings for Configuration Data Lock Bit (CD<sub>n</sub>\_LK) (n = 0 to 2) are shown in [Table 6.4](#)

**Table 6.4 The correspondence between the lock bit and lockable are address (1 of 2)**

Area	Bit	Address
Lockable area 0	CD0_LK6	0x2703_0098 to 0x2703_009B
	CD0_LK7	0x2703_009C to 0x2703_009F
	...	...
	CD0_LK <sub>n</sub> (n = 6 to 63)	0x2703_0080 + 0x04 × n to 0x2703_0083 + 0x04 × n
	...	...
	CD0_LK63	0x2703_017C to 0x2703_017F
Lockable area 1	CD1_LK0	0x2703_0180 to 0x2703_0183
	CD1_LK1	0x2703_0184 to 0x2703_0187
	...	...
	CD1_LK <sub>n</sub> (n = 0 to 63)	0x2703_0180 + 0x04 × n to 0x2703_0183 + 0x04 × n
	...	...
	CD1_LK63	0x2703_00x27C to 0x2703_027F

**Table 6.4 The correspondence between the lock bit and lockable are address (2 of 2)**

Area	Bit	Address
Lockable area 2	CD2_LK0	0x2703_0280 to 0x2703_028F
	CD2_LK1	0x2703_0290 to 0x2703_029F
	...	...
	CD2_LKn (n = 0 to 13)	0x2703_0280 + 0x10 × n to 0x2703_028F + 0x10 × n
	...	...
	CD2_LK13	0x2703_0350 to 0x2703_035F

### 6.2.14 ARCLS : Anti-Rollback Counter Lock Setting

Base address: 0x2703\_03C0

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARCNS_LK[3:0]	ARCS_LK
------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---------------	---------

Value after reset: User setting\*1

Bit	Symbol	Function	R/W
0	ARCS_LK	ARC_SEC Lock 0: ARC_SEC protection is enabled 1: ARC_SEC protection is disabled	R/W
4:1	ARCNS_LK[3:0]	ARC_NSEC Lock In case of 64 bits x 4 counter setting x x x 0: ARC_NSEC[63:0] protection is enabled x x 0 x: ARC_NSEC[127:64] protection is enabled x 0 x x: ARC_NSEC[191:128] protection is enabled 0 x x x: ARC_NSEC[255:192] protection is enabled 1 1 1 1: ARC_NSEC[255:0] protection is disabled Others Reserved. In case of 256 bits counter setting 0 0 0 0: ARC_NSEC[255:0] protection is enabled 1 1 1 1: ARC_NSEC[255:0] protection is disabled Others Reserved.	R/W
15:5	—	The program value is read from these bits*2	R/W

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application

Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

Anti-rollback counter Lock bits setting are located in the data flash configuration area. Anti-rollback counter Lock bits setting can be set to "0" by the "Program" command and cannot be changed from "0" to "1". When the lock bit is 0, the anti-rollback counter corresponding to each bit cannot be updated by the "Increment Counter" command.

### 6.2.15 ARCCS : Anti-Rollback Counter Configuration Setting for Non-secure Application

Base address: 0x2703\_03C2

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CNF_ARCNS[1:0]
------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----------------

Value after reset: User setting\*1

Bit	Symbol	Function	R/W
1:0	CNF_ARCNS[1:0]	Configuration setting for ARC_NSEC 0 0: ARC_NSEC configuration is 4 × 64 bits 0 1: ARC_NSEC configuration is 1 × 256 bits Others ARC_NSEC configuration is disable	R/W

Bit	Symbol	Function	R/W
15:2	—	The program value is read from these bits*2	R/W

Note 1. The value in a blank product is 0xFFFFFFFF. It is set to the value written by your application

Note 2. This register can only be programmed by FACI command. When programming, the set value should be 1.

ARC\_NSEC configuration setting is located in the data flash configuration area. These bits specify two types of ARC\_NSEC configuration:  $4 \times 64$  bits or  $1 \times 256$  bits. “Increment Counter” or “Read Counter” command cannot be issued to ARC\_NSEC when counter configuration setting is disabled.

### 6.2.16 ARC\_SECN : Anti-Rollback Counter for Secure Application n (n = 0, 1)

Base address: 0x2703\_0000

Offset address:  $0x850 + 0x004 \times n$  (n = 0, 1)

Bit position: 31

0

Bit field:

ARC\_SECN[32 × n + 31 : 32 × n]

Value after reset:

User setting\*1

Bit	Symbol	Function	R/W
31:0	ARC_SECN[32 × n + 31 : 32 × n]	Anti-Rollback Counter for Secure Application The counter value is obtained by arranging the read values from the upper register (n = 1) to the lower register (n = 0). See <a href="#">section 49.12.5. Anti-Rollback Counter</a> for detail	R/W

Note 1. The value in a blank product is 0x0. It is set to the value written by your application

### 6.2.17 ARC\_NSECn : Anti-Rollback Counter for Non-secure Application n (n = 0 to 7)

Base address: 0x2703\_0000

Offset address:  $0x858 + 0x004 \times n$  (n = 0 to 7)

Bit position: 31

0

Bit field:

ARC\_NSECn[32 × n + 31 : 32 × n]

Value after reset:

User setting\*1

Bit	Symbol	Function	R/W
31:0	ARC_NSECn[32 × n + 31 : 32 × n]	Anti-Rollback Counter for Non-secure Application The counter value is obtained by arranging the read values from the upper register (n = 7) to the lower register (n = 0). See <a href="#">section 49.12.5. Anti-Rollback Counter</a> for detail	R/W

Note 1. The value in a blank product is 0x0. It is set to the value written by your application

## 6.3 Setting Option-Setting Memory

### 6.3.1 Allocation of Data in Option-Setting Memory

Programming data is allocated to the addresses in the option-setting memory shown in [Figure 6.1](#). The allocated data is used by tools such as a flash programming software or an on-chip debugger.

Note: Programming formats vary depending on the compiler. See the compiler manual for details.

### 6.3.2 Setting Data for Programming Option-Setting Memory

Allocating data according to the procedure described in [section 6.3.1. Allocation of Data in Option-Setting Memory](#), alone does not actually write the data to the option-setting memory. You must also follow one of the actions described in this section.



### (1) Changing the option-setting memory by self-programming

Use the configuration setting command to write data to the option-setting memory in the configuration setting area.

The code flash option-setting memory does not support background operations (BGO). When write the option-setting memory, jump to SRAM after copying writing software to SRAM.

For details of the configuration setting command, see [section 49, Flash Memory](#).

### (2) Debugging through an OCD or programming by a flash writer

This procedure depends on the tool in use, see the tool manual for details.

The MCU provides two setting procedures:

- Read the data allocated as described in [section 6.3.1. Allocation of Data in Option-Setting Memory](#), from an object file or Motorola S-format file generated by the compiler, and write the data to the MCU
- Use the GUI interface of the tool to program the same data as allocated in [section 6.3.1. Allocation of Data in Option-Setting Memory](#).

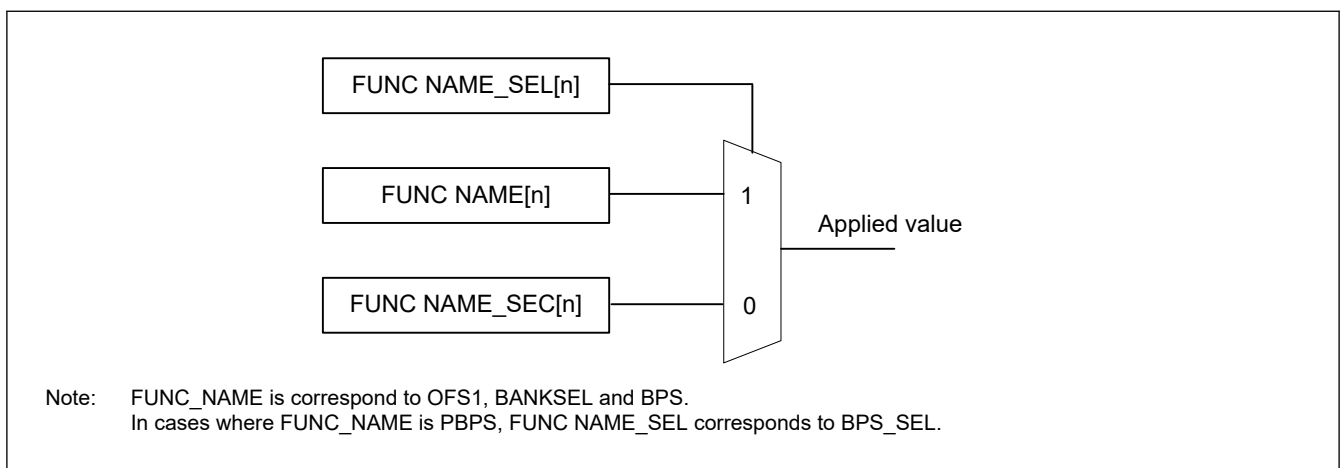
## 6.3.3 Security attribution of code flash option-setting memory

Some functionality has 3 registers for Non-secure (FUNC NAME), and Secure (FUNC NAME\_SEC), and security attribution (FUNC NAME\_SEL). Only Secure developer can set the registers for Secure and security attribution. As shown in [Figure 6.5](#), when the bit of security attribution register is set to 0, the corresponding bit of Secure register is applied. When the bit of security attribution register is set to 1, the corresponding bit of Non-secure register is applied.

For example, if the Secure developer wants to configure PVD of OFS1 as Secure, HOCO of OFS1 as Non-secure, the Secure developer needs to set OFS1\_SEL as follows.

OFS1\_SEL = 0xFFFF\_FFF0

By this setting, PVDAS and VDSEL[2:0] values of OFS1\_SEC and HOCOFrq0[2:0] and HOCOEN values of OFS1 are applied to MCU. The reserved bits of the security attribution register (FUNC NAME\_SEL) should be set to 0.



**Figure 6.5 Selection of applied value**

### 6.3.4 Timing of the Setting Value

For SAS, BPS, BPS\_SEC, PBPS, PBPS\_SEC and BPS\_SEL registers, the setting value of the related startup area and block protection is applied immediately after programming. For other registers, the setting value is applied after the MCU is reset.

## 6.4 Usage Notes

### 6.4.1 Data for Programming Reserved Areas and Reserved Bits in the Option-Setting Memory

- Security attribution region (0x0300 A280 to 0x0300 A2FF)

When reserved areas and reserved bits in the option-setting memory are within the scope of programming, write 0 to all bits in reserved areas and all reserved bits. Operation is not guaranteed if 1 is written to these bits.

- Other region

When reserved areas and reserved bits in the option-setting memory are within the scope of programming, write 1 to all bits of reserved areas and all reserved bits. If 0 is written to these bits, normal operation cannot be guaranteed.

## 7. Programmable Voltage Detection (PVD)

### 7.1 Overview

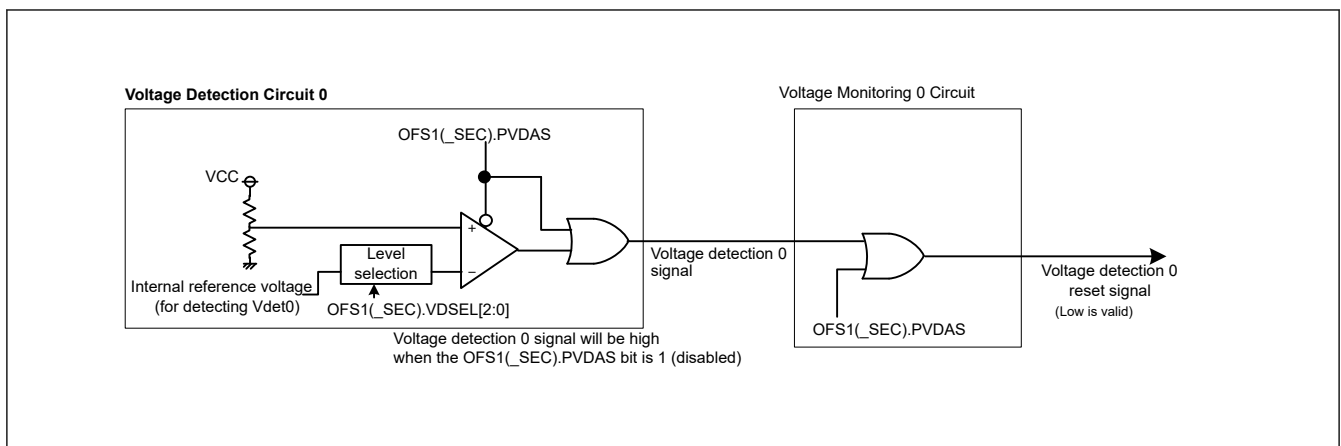
The Programmable Voltage Detection (PVD) module monitors the voltage level input to the VCC pin and the detection level can be selected using a software program. The PVD module consists of several separate voltage level detectors which measure the voltage level input to the VCC pin. PVD registers allow your application to configure detection of VCC change at various voltage thresholds.

Each voltage level detector has a voltage monitor associated with it. Voltage monitor registers are used to configure the PVD to trigger an interrupt, event link output, or reset when the thresholds are crossed.

Table 7.1 lists the PVD specifications. Figure 7.1 shows a block diagram of the voltage monitor 0 reset generation circuit. Figure 7.2 shows a block diagram of the voltage monitor m interrupt and reset circuit (m = 1, 2).

**Table 7.1 PVD specifications**

Item		Voltage monitor 0	Voltage monitor m (m = 1, 2)
Voltage Monitoring	Monitored pin	VCC	VCC
	Detected Event	Voltage falls past $V_{det0}$	Voltage rises or falls past $V_{detm}$
	Detection Voltage	Selectable from among 8 different levels by using OFS1(_SEC).VDSEL[2:0] bits as $V_{det0}$	Selectable from among 13 different levels by using PVDmCMPCR.PVDLVL[4:0] bits as $V_{detm}$
	Monitor Flag	None	PVDmSR.MON flag: Monitors whether voltage is higher or lower than $V_{detm}$ PVDmSR.DET flag: $V_{detm}$ passage detection
Process on voltage detection	Reset	Voltage monitor 0 reset	Voltage monitor m reset
		Reset when $V_{det0} > VCC$ CPU restart after specified time with $VCC > V_{det0}$	Reset condition selectable Reset when $V_{detm} > VCC$ or $V_{detm} < VCC$ CPU restart timing selectable: after specified time with $VCC > V_{detm}$ or $V_{detm} > VCC$
Interrupt	No interrupt	No interrupt	Voltage monitor m interrupt
			Non-maskable interrupt or maskable interrupt selectable
			Interrupt request issued in response to either or both $V_{detm} > VCC$ and $VCC > V_{detm}$
Digital filter	Enable/ Disable Switching	Digital filter function not available	Available
	Sampling Time	—	1/n LOCO frequency × 2 (n: 2, 4, 8, 16)
Event linking		None	Available Output of event signals on detection of $V_{detm}$ crossings



**Figure 7.1 Block diagram of voltage monitor 0 reset circuit**

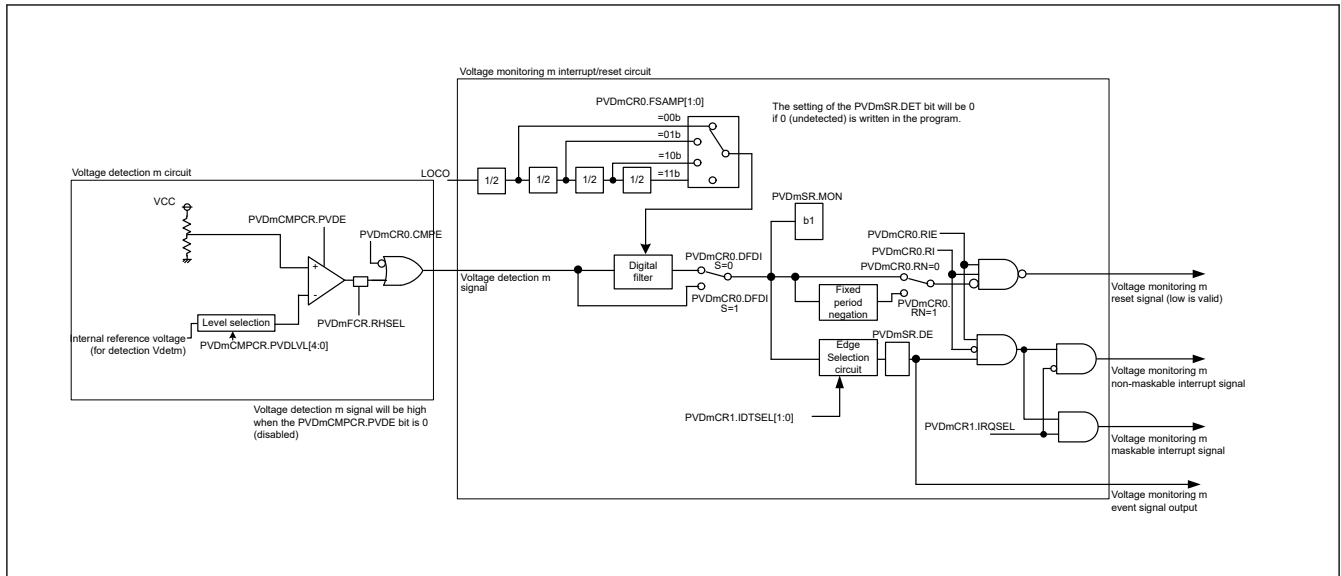


Figure 7.2 Block diagram of voltage monitor m interrupt/reset circuit (m = 1, 2)

## 7.2 Register Descriptions

### 7.2.1 PVDSAR : Programable Voltage Detection Security Attribution Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x3CC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NONSEC1	NONSEC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non Secure Attribute bit 0 Target register: registers for PVD1 0: Secure 1: Non Secure	R/W
1	NONSEC1	Non Secure Attribute bit 1 Target register: registers for PVD2 0: Secure 1: Non Secure	R/W
31:2	—	These bits are read as 1. The write value must be 1 when it is possible to write.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: This register is write-protected by PRCR register.

The PVDSAR register controls the secure attribute of PVD registers.

#### NONSEC0 bit (Non Secure Attribute bit 0)

This bit controls the security attribute of PVD1CMPCR, PVD1CR0, PVD1CR1, PVD1SR and PVD1FCR.

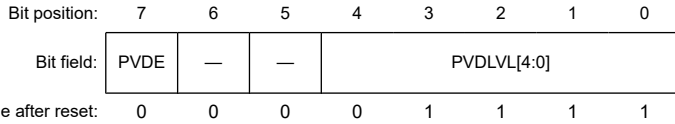
#### NONSEC1 bit (Non Secure Attribute bit 1)

This bit controls the security attribute of PVD2CMPCR, PVD2CR0, PVD2CR1, PVD2SR and PVD2FCR.

### 7.2.2 PVDmCMPCR : Voltage Monitor m Comparator Control Register (m = 1, 2)

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0xA58 + 0x4 × (m - 1)



Bit	Symbol	Function	R/W
4:0	PVDLVL[4:0]	Detection Voltage m Level Select Settings other than the following are prohibited. These are standard voltage level during the fall of voltage. For details on standard voltage level during the rise of voltage, see <a href="#">section 52, Electrical Characteristics</a> . 0x03: 3.86 V (Vdetm_3) 0x04: 3.14 V (Vdetm_4) 0x05: 3.10 V (Vdetm_5) 0x06: 3.08 V (Vdetm_6) 0x07: 2.85 V (Vdetm_7) 0x08: 2.83 V (Vdetm_8) 0x09: 2.80 V (Vdetm_9) 0x0A: 2.62V (Vdetm_10) 0x0B: 2.33V (Vdetm_11) 0x0C: 1.90V (Vdetm_12) 0x0D: 1.86V (Vdetm_13) 0x0E: 1.74V (Vdetm_14) 0x0F: 1.71V (Vdetm_15) Others: Setting prohibited	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	PVDE	Voltage Detection m Enable 0: Voltage detection m circuit disabled 1: Voltage detection m circuit enabled	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

#### PVDLVL[4:0] bits (Detection Voltage m Level Select)

The PVDmCMPCR.PVDLVL can be changed only if the PVDmCMPCR.PVDE bits are both 0. All the voltage detection circuits should not be set at the same voltage detection level.

Do not change PVDmCMPCR.PVDLVL and PVDmCMPCR.PVDE at the same time.

The PVDmCMPCR.PVDE can be changed only when the PVDmCR0.CMPE is 0.

#### PVDE bit (Voltage Detection m Enable)

When using voltage detection m interrupt/reset or the PVDmSR.MON bit, set the PVDmCMPCR.PVDE bit to 1. The voltage detection m circuit starts once td(E-A) passes after the PVDmCMPCR.PVDE bit value is changed from 0 to 1.

When using the voltage detection m circuit in Deep Software Standby mode, do not transit to Deep Software Standby mode 3.

### 7.2.3 PVDmCR0 : Voltage Monitor m Circuit Control Register 0 (m = 1, 2)

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA70 + 0x4 × (m - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RN	RI	FSAMP[1:0]	—	CMPE	DFDIS	RIE	
Value after reset:	1	0	0	0	x	0	1	0

Bit	Symbol	Function	R/W
0	RIE	Voltage Monitor m Interrupt/Reset Enable 0: Disable 1: Enable	R/W
1	DFDIS	Voltage monitor m Digital Filter Disabled Mode Select 0: Enable the digital filter 1: Disable the digital filter	R/W
2	CMPE	Voltage Monitor m Circuit Comparison Result Output Enable 0: Voltage monitor m circuit comparison result output disabled 1: Voltage monitor m circuit comparison result output enabled	R/W
3	—	The read value is undefined. The write value should be 1.	R/W
5:4	FSAMP[1:0]	Sampling Clock Select 0 0: 1/2 LOCO frequency 0 1: 1/4 LOCO frequency 1 0: 1/8 LOCO frequency 1 1: 1/16 LOCO frequency	R/W
6	RI	Voltage Monitor m Circuit Mode Select (1) case of PVDmFCR.RHSEL = 0 0: Voltage monitor m interrupt during $V_{detm}$ passage 1: Voltage monitor m reset enabled when the voltage falls to and below $V_{detm}$ (2) case of PVDmFCR.RHSEL = 1 0: prohibited 1: Voltage monitoring m reset enabled when the voltage rises to and up $V_{detm}$	R/W
7	RN	Voltage Monitor m Reset Negate Select (1) case of PVDmFCR.RHSEL = 0 0: Negation follows a stabilization time ( $t_{pVDm}$ ) after $VCC > V_{detm}$ is detected 1: Negation follows a stabilization time ( $t_{pVDm}$ ) after assertion of the PVDm reset. (2) case of PVDmFCR.RHSEL = 1 0: Negation follows a stabilization time ( $t_{pVDm}$ ) after $VCC < V_{detm}$ is detected. 1: Prohibited	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

#### RIE bit (Voltage Monitor m Interrupt/Reset Enable)

The RIE bit enables or disables voltage monitor m interrupt/reset. Ensure that neither a voltage monitor m interrupt nor a voltage monitor m reset is generated during programming or erasure of the flash memory.

#### DFDIS bit (Voltage monitor m Digital Filter Disabled Mode Select)

The DFDIS bit disables the digital filter circuit. Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if the DFDIS bit is 0 (enabling the digital filter circuit). Set the DFDIS bit to 1 (digital filter circuit disabled) when using voltage monitor m circuit in software standby mode or Deep Software Standby mode.

#### CMPE bit (Voltage Monitor m Circuit Comparison Result Output Enable)

The CMPE bit enables or disables voltage monitor m circuit comparison result output. Set the CMPE bit to 1 after the voltage detection m circuit enables and stabilization time ( $t_{d(E-A)}$ ) elapses. When stopping the voltage detection m circuit, disable the voltage detection m circuit after setting the CMPE bit is 0.

**FSAMP[1:0] bits (Sampling Clock Select)**

The FSAMP[1:0] bits can be rewritten only when the PVDmCR0.DFDIS bit is 1 (digital filter circuit disabled). The FSAMP[1:0] bits should not be modified when the PVDmCR0.DFDIS bit is 0 (digital filter circuit enabled).

**RI bit (Voltage Monitor m Circuit Mode Select)**

When the PVDmCR0.RI bit is 1 (voltage monitor m reset selected), a transition to Deep Software Standby mode 2 or 3 cannot be made, instead a transition to Deep Software Standby mode 1 is made. To enter Deep Software Standby mode 2 or 3, set all the PVDmCR0.RI bits to 0 (voltage monitor m interrupt selected).

**RN bit (Voltage Monitor m Reset Negate Select)**

Case of PVDmFCR.RHSEL = 0

- If the RN bit is set to 1 (negation follows a stabilization time after assertion of the PVDm reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to Software Standby mode or Deep Software Standby mode is to be made, the only possible value for the RN bit is 0 (negation follows a stabilization time after  $VCC > V_{detm}$  is detected. Do not set the RN bit to 1 (negation follows a stabilization time after assertion of the PVDm reset signal) when this is the case.

Case of PVDmFCR.RHSEL = 1

- Do not set the RN bit to 1.

**7.2.4 PVDmCR1 : Voltage Monitor m Circuit Control Register 1 (m = 1, 2)**

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x0E0 + 0x2 × (m-1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IRQSEL	IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
1:0	IDTSEL[1:0]	Voltage Monitor m Interrupt Generation Condition Select (1) case of PVDmFCR.RHSEL = 0 0 0: When $VCC \geq V_{detm}$ (rise) is detected 0 1: When $VCC < V_{detm}$ (fall) is detected 1 0: When fall and rise are detected 1 1: Settings prohibited (2) case of PVDmFCR.RHSEL = 1 This bit is not applicable.	R/W
2	IRQSEL	Voltage Monitor m Interrupt Type Select 0: Non-maskable interrupt 1: Maskable interrupt*1	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC3 bit to 1 (writing enabled) before rewriting this register.

Note 1. When enabling maskable interrupts, do not change the value of the NMIER.PVDmEN bit on the ICU side from the reset state.

### 7.2.5 PVDmSR : Voltage Monitor m Circuit Status Register (m = 1, 2)

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x0E1 + 0x2 × (m - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MON	DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	DET	Voltage Monitor m Voltage Change Detection Flag* <sup>1</sup> (1) case of PVDmFCR.RHSEL = 0 0: Not detected 1: $V_{detm}$ passage detection (2) case of PVDmFCR.RHSEL = 1 This bit is not applicable.	R/W
1	MON	Voltage Monitor 1 Signal Monitor Flag (1) case of PVDmFCR.RHSEL = 0 0: $VCC \leq V_{detm}$ 1: $VCC > V_{detm}$ or MON is disabled (2) case of PVDmFCR.RHSEL = 1 This bit is not applicable.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

Note 1. Only 0 can be written to this bit. After writing 0 to this bit, 2 system clock (ICLK) cycles are required for the bit to be read as 0.

#### DET flag (Voltage Monitor m Voltage Change Detection Flag)

The DET flag is enabled when the PVDmCMPCR.PVDE bit is 1 (voltage detection m circuit enabled) and the PVDmCR0.CMPE bit is 1 (voltage monitor m circuit comparison result output enabled).

The DET flag should be set to 0 after PVDmCR0.RIE is set to 0 (disabled). PVDmCR0.RIE can be set to 1 (enabled) after a period of two or more cycles of PCLKB has elapsed.

Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles than PCLKB may have to be secured as waiting time.

#### MON flag (Voltage Monitor 1 Signal Monitor Flag)

The MON flag is enabled when the PVDmCMPCR.PVDE bit is 1 (voltage detection m circuit enabled) and the PVDmCR0.CMPE bit is 1 (voltage monitor m circuit comparison result output enabled).

### 7.2.6 PVDmFCR : Voltage Monitor m Function Control Register (m = 1, 2)

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xB20 + 0x4 × (m - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RHSEL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RHSEL	Rise Hysteresis Select 0: Hysteresis level for VCC-fall detection is selected. 1: Hysteresis level for VCC-rise detection is selected.	R/W



Bit	Symbol	Function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

The PVDmFCR.RHSEL can be changed only when all the PVDmCMPCR.PVDE bits are 0.

### RHSEL flag (Rise Hysteresis Select)

When RHSEL=0 for VCC-fall detection, the hysteresis level is set above the voltage detection level set by PVDLVL[4:0]. When RHSEL=1 for VCC-rise detection, the hysteresis level is set under the voltage detection level set by PVDLVL[4:0]. Refer to the chapter of electrical characteristics about the detail of hysteresis level of PVDm.

When PVDm reset generated by the VCC-fall detection is required, set RHSEL to 0.

When PVDm reset generated by the VCC-rise detection is required, set RHSEL to 1.

RHSEL must not be set to “1” when PVDmCR0.RI = 0.

RHSEL must not be set to “1” when PVDmCR0.RN = 1.

## 7.3 VCC Input Voltage Monitor

### 7.3.1 Monitoring $V_{det0}$

The comparison results from voltage monitor 0 are not available for reading.

### 7.3.2 Monitoring $V_{detm}$

After the settings are completed, results of comparison by voltage Monitor m can be monitored by using the PVDmSR.MON flag.

**Table 7.2 Procedures to set up monitoring against  $V_{detm}$**

Step	Monitoring the comparison results from voltage monitor 1	
Setting the voltage detection m circuit	1	Set PVDmCMPCR.PVDE = 0 to disable voltage detection m before writing to the PVDmCMPCR.PVDLVL[4:0] bits.
	2	Select the detection voltage in the PVDmCMPCR.PVDLVL[4:0] bits.
	3	Set PVDmCMPCR.PVDE = 1 to enable the voltage detection m circuit.
	4	Wait for at least $t_d (E-A)^{-1}$
Setting the digital filter <sup>2</sup>	5	Select the sampling clock for the digital filter in the PVDmCR0.FSAMP[1:0] bits.
	6	Set PVDmCR0.DFDIS = 0 to enable the digital filter.
	7	Wait for at least $2n + 3$ cycles of the LOCO (where $n = 2, 4, 8, 16$ , and the sampling clock for the digital filter is the LOCO frequency-divided by $n$ ).
Enabling output	8	Set PVDmCR0.CMPE = 1 to enable output of the comparison results from voltage monitor m.

Note 1. Steps 5 to 7 can be performed during the waiting time of step 4.

Note 2. Steps 5 to 7 are not required if the digital filter is not in use.

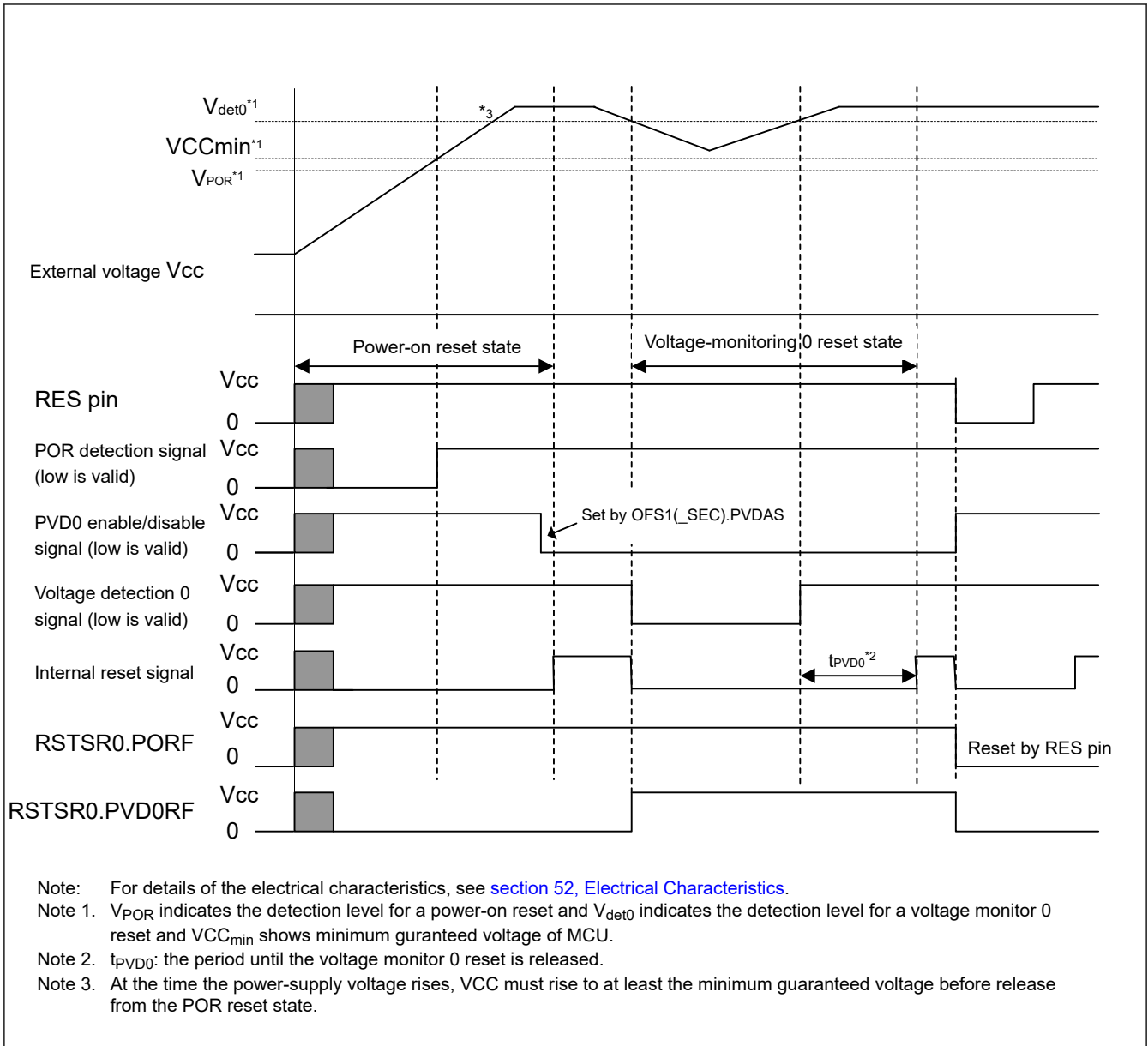
After configuring the PVDm according to the procedure in the table above, if you change the PVDm setting again, first set PVDmCR0.CMPE to 0, and then configure PVDm again according to the procedure in the table above.

When changing the PVDmCMPCR.PVDLVL, comply with the description of the [section 7.2.2. PVDmCMPCR : Voltage Monitor m Comparator Control Register \(m = 1, 2\)](#).

## 7.4 Reset from Voltage Monitor 0

When using the reset from voltage monitor 0, clear the OFS1(\_SEC).PVDAS bit to 0. (enabling the voltage monitor 0 reset after a reset)

Figure 7.3 shows an example of operations for a voltage monitor 0 reset.



**Figure 7.3 Example of voltage monitor 0 reset operation**

### 7.5 Interrupt and Reset from Voltage Monitor m (m = 1, 2)

An interrupt or reset can be generated in response to the results of comparison by the voltage detection m circuit

[Table 7.3](#) shows the procedures for setting bits related to the voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring operates. [Table 7.4](#) shows the procedures for setting bits related to the voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring stops. [Figure 7.4](#) shows an example of operations for a voltage monitor m interrupt. For the operation of the voltage monitor m reset, see [Figure 5.2](#) in [section 5, Resets](#).

Furthermore, if you intend to use the voltage Monitor m circuit in software standby or Deep Software Standby mode, make settings for the voltage Monitor m circuit according to the following procedures.

#### (1) Setting in Software Standby mode

- Disable the digital filter ( $PVDmCR0.DFDIS = 1$ ).
- Case of  $PVDmFCR.RHSEL = 0$
- After  $V_{CC} > V_{detm}$  is detected, negate the voltage monitor m reset signal ( $PVDmCR0.RN = 0$ ) following a stabilization time.

## (2) Settings in Deep Software Standby mode

- Disable the digital filter (PVDmCR0.DFDIS = 1).
- Enable the voltage monitor m interrupt (PVDmCR0.RI = 0). If the voltage monitor m reset is enabled (PVDmCR0.RI = 1), a transition to Deep Software Standby mode 2 or 3 is not be possible, and the transition will be to Deep Software Standby mode 1 instead.
- During Deep Software Standby mode 3, the voltage Monitor m circuit is stopped. If you intend to use the voltage Monitor m circuit in Deep Software Standby mode, do not transit to Deep Software Standby mode 3.

## (3) Disabling at on-chip debug mode

- The function of the voltage Monitor m circuit is disabled by setting DBGSTOPCR.DBGSTOP\_PVD at on-chip debug mode.

**Table 7.3 Procedures for setting bits related to voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring operates (m = 1, 2)**

Step	Voltage monitor m interrupt (voltage monitor m ELC event output)	Voltage monitor m reset	
Setting up the voltage detection m circuit	1	Set PVDmCMPCR.PVDE = 0 (disable voltage detection m before writing to the PVDmCMPCR.PVDLVL and PVDmFCR.RHSEL register.)	
	2	Select the detection voltage by setting the PVDmCMPCR.PVDLVL bits.	
	3	—	Select the type of the reset condition by setting the PVDmFCR.RHSEL bit.
	4	Set PVDmCMPCR.PVDE = 1 (enabling the voltage detection m circuit).	
	5	Wait for at least $t_d$ (E-A). *1	
Setting the digital filter*2	6	Select the sampling clock for the digital filter by the setting PVDmCR0.FSAMP[1:0] bits.	
	7	Set PVDmCR0.DFDIS = 0 (enabling the digital filter).	
	8	Wait for at least $2n + 3$ cycles of the LOCO (where $n = 2, 4, 8, 16$ , and the sampling clock for the digital filter is the LOCO frequency-divided by $n$ ). *4.	
Setting up the voltage monitor m interrupt or reset	9	Set PVDmCR0.RI = 0 to select the voltage monitor m interrupt.	<ul style="list-style-type: none"> <li>• Set PVDmCR0.RI = 1 (selecting the voltage monitor m reset).</li> <li>• Select the type of the reset negation by setting the PVDmCR0.RN bit.</li> </ul>
	10	<ul style="list-style-type: none"> <li>• Select the timing of interrupt requests by setting the PVDmCR1.IDTSEL[1:0] bits.</li> <li>• Select the type of interrupt by setting the PVDmCR1.IRQSEL bit.</li> </ul>	—
Enabling output	11	Set PVDmSR.DET = 0.	
	12	Set PVDmCR0.RIE = 1 (enabling the voltage Monitor m interrupt or reset). *3	
	13	Set PVDmCR0.CMPE = 1 (enabling output of the results of comparison by voltage monitor m).	

Note 1. Steps 6 to 12 can be performed during the wait time in step 4.

Note 2. Steps 6 to 8 are not required if the digital filter is not in use.

Note 3. Step 12 is not required if only the ELC event signal is to be output.

Note 4. Steps 9 to 12 can be performed during the waiting time of step 8.

After configuring the PVDm according to the procedure in the [Table 7.3](#), if you change the PVDm settings again, stop PVDm according to the procedure in [Table 7.4](#), and then configure PVDm again according to the procedure in the [Table 7.3](#).

When changing the PVDmCMPCR.PVDLVL, comply with the description of the [section 7.2.2. PVDmCMPCR : Voltage Monitor m Comparator Control Register \(m = 1, 2\)](#).

**Table 7.4 Procedures for setting bits related to voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring stops (m = 1, 2) (1 of 2)**

Step	Voltage monitor m interrupt (voltage monitor m ELC event output), voltage monitor m reset
Stopping the enabling output	1 Set PVDmCR0.CMPE = 0 (disabling output of the results of comparison by voltage Monitor m).

**Table 7.4 Procedures for setting bits related to voltage monitor m interrupt and voltage monitor m reset so that voltage monitoring stops (m = 1, 2) (2 of 2)**

Step	Voltage monitor m interrupt (voltage monitor m ELC event output), voltage monitor m reset	
	2	Wait for at least $2n + 3$ cycles of the LOCO (where $n = 2, 4, 8, 16$ , and the sampling clock for the digital filter is the LOCO frequency-divided by $n$ .) <sup>*1</sup>
	3	Set PVDmCR0.RIE = 0 (disabling the voltage monitor m interrupt or reset). <sup>*3</sup>
Stopping the digital filter	4	Set PVDmCMPCR.DFDIS = 1 (disabling the digital filter). <sup>*1, *2</sup>
Stopping the voltage detection m circuit	5	Set PVDmCMPCR.PVDE = 0 (disabling the voltage detection m circuit).

Note 1. Steps 2 and 4 are not required if the digital filter is not in use.

Note 2. To disable the digital filter from its enabled state and then re-enable it, disable it and wait for at least two cycles of the LOCO before re-enabling it.

Note 3. Step 3 is not required if only the ELC event signal is to be output.

If the voltage monitor m interrupt or voltage monitor m reset setting is to be made again after it has been used and stopped once, the following steps in the procedures for stopping and making the setting can be omitted according to the condition:

- Setting or stopping the voltage detection m circuit is not required if the setting for the voltage detection m circuit is not to be changed
- Setting or stopping the digital filter is not required if the setting for the digital filter is not to be changed
- Setting the voltage monitor m interrupt or reset is not required if the setting for the voltage monitor m interrupt or voltage monitor m reset is not to be changed.

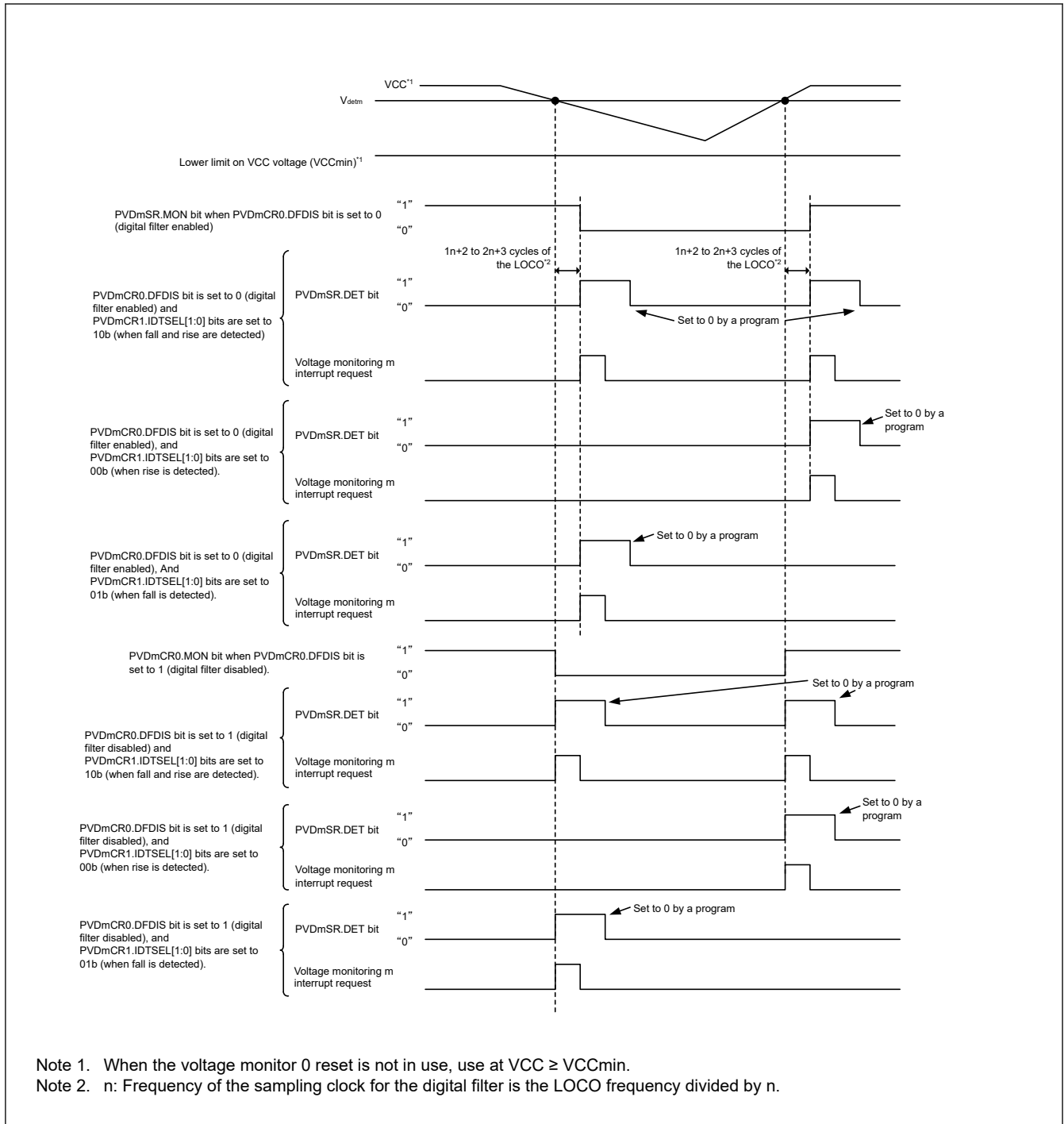


Figure 7.4 Example of voltage monitor m interrupt operation (m = 1, 2)

## 7.6 Event Link Controller (ELC) Output

The PVDm (m = 1, 2) can output the event signals to the Event Link Controller (ELC).

### (1) V<sub>detm</sub> passage detection event

The PVDm outputs the event signal when it is detected that the voltage has passed the V<sub>detm</sub> voltage while both the voltage detection m circuit and the voltage monitor m circuit comparison result output are enabled.

When enabling the event link output function of the PVDm, be sure to specify settings to enable the PVDm before enabling the PVDm event link function of the ELC. To stop the event link output function of the PVDm, be sure to specify settings to stop the PVDm before disabling the PVDm event link function of the ELC.

### 7.6.1 Interrupt Handling and Event Linking

Each PVD has bits to separately enable or disable the voltage monitor m interrupts. When an interrupt source is generated and the interrupt is enabled by the interrupt enable bit, the interrupt signal is output to the CPU.

On the contrary, as soon as an interrupt source is generated, the event link signal is output as the event signal to the other module through the ELC regardless of the state of the interrupt enable bit.

It is possible to output voltage monitor m interrupts in Software Standby mode, Deep Software Standby mode 1, and Deep Software Standby mode 2. The event signals for the ELC in Software Standby mode, Deep Software Standby mode 1, and Deep Software Standby mode 2 are output as follows:

- When the event  $V_{detm}$  is detected in Software Standby mode, no event signals are generated for the ELC because no clock is presented in Software Standby mode. Because  $V_{detm}$  passage detection flag is preserved, when the supply of the clock is resumed after restoring from Software Standby mode, the event signals for the ELC is output according to the state of the  $V_{detm}$  passage detection flag.
- If event of passing  $V_{detm}$  is detected in Deep Software Standby mode, no event signal is generated for the ELC.

## 8. Clock Generation Circuit

### 8.1 Overview

The MCU provides a clock generation circuit. [Table 8.1](#) and [Table 8.2](#) list the clock generation circuit specifications. [Figure 8.1](#) show a block diagram, and [Table 8.3](#) lists the I/O pins.

**Table 8.1 Clock generation circuit specifications for the clock sources (1 of 2)**

Clock source	Description	Specification
Main clock oscillator (MOSC)	Resonator frequency	8 MHz to 48 MHz (For frequencies available in USB boot mode, see the application note of the boot firmware)
	External clock input frequency	Up to 48 MHz
	Connectable resonator or additional circuit: ceramic resonator, crystal Connection pins: EXTAL, XTAL	Available
	Drive capability switching	Available
	Automatic Gain Control Function	Available
	Oscillation stop detection function	Available
Sub-clock oscillator (SOSC)	Resonator frequency	32.768 kHz
	Connectable resonator or additional circuit: crystal resonator Connection pin: XCIN, XCOU	Available
	External clock input	Available
	Drive capability switching	Available
PLL1 circuit PLL2 circuit	Input clock source	MOSC/HOCO
	Input pulse frequency division ratio	Selectable from 1/2/3/4
	Input clock frequency	8 MHz to 48 MHz
	Input clock frequency (After input frequency division)	6 MHz to 12 MHz
	Frequency multiplication ratio	Selectable from 53 to 180 (after the decimal point : 0/0.33/0.50/0.66)
	VCO frequency	640 MHz to 1440 MHz
	Number of output clocks	Output 3 different clocks
	PLL Output clock P	40 MHz to 480 MHz (output division ratio : 2/4/6/8/16)
	PLL Output clock Q	71 MHz to 480 MHz (output division ratio : 2/3/4/5/6/8/9)
PLL Output clock R	71 MHz to 480 MHz (output division ratio : 2/3/4/5/6/8/9)	
High-speed on-chip oscillator (HOCO)	Oscillation frequency	16/18/20/32/48 MHz
	FLL function	Available
	User trimming	Available
Middle-speed on-chip oscillator (MOCO)	Oscillation frequency	8 MHz
	User trimming	Available
Low-speed on-chip oscillator (LOCO)	Oscillation frequency	32.768 kHz
	User trimming	Available
External clock input for JTAG (TCK)	Input clock frequency	Up to 25 MHz

**Table 8.1 Clock generation circuit specifications for the clock sources (2 of 2)**

Clock source	Description	Specification
External clock input for SWD (SWCLK)	Input clock frequency	Up to 25 MHz

**Table 8.2 Clock generation circuit specifications for the internal clocks (1 of 2)**

Item	Clock source	Clock supply	Specification
CPU clock (CPUCLK)	MOSC/SOSC/HOCO/MOCO/PLL1P	CPU, TCM	Up to 360 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
System clock (ICLK)	MOSC/SOSC/HOCO/MOCO/PLL1P	DMAC, DTC, Flash (Reading from code flash), SRAM, Standby SRAM, System BUS, I/O Ports, ICU	Up to 240 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
Debugger clock (DCLK)	1/2 ICLK	Debug Subsystem	Up to 120 MHz
Peripheral module clock A (PCLKA)	MOSC/SOSC/HOCO/MOCO/PLL1P	High-speed peripheral bus (ETHERC, EDMAC, SCI, CANFD, CNECC, SPI, CRC, DOC, ADC12, DAC12, RSIP-E51A, GPT, CEU)	Up to 120 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
Peripheral module clock B (PCLKB)	MOSC/SOSC/HOCO/MOCO/PLL1P	Low-speed peripheral bus (CAC, ELC, POEG, RTC, WDT, IWDT, AGT, ULPT, IIC, USBFS, SSIE, TSN, ACMPHS, OSPI)	Up to 60 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
Peripheral module clock C (PCLKC)	MOSC/SOSC/HOCO/MOCO/PLL1P	Peripheral module (ADC12 conversion clock)	Up to 60 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
Peripheral module clock D (PCLKD)	MOSC/SOSC/HOCO/MOCO/PLL1P	Peripheral module(GPT count clock)	Up to 120 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
Peripheral module clock E (PCLKE)	MOSC/SOSC/HOCO/MOCO/PLL1P	Peripheral module (CNECC)	Up to 240 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64
FlashIF clock (FCLK)	MOSC/SOSC/HOCO/MOCO/PLL1P	FlashIF	4 MHz to 60 MHz (P/E) Up to 60 MHz (read) Division ratio: 1/2/3/4/6/8/12/16/32/64
Trace clock (TRCLK)	MOSC/SOSC/HOCO/MOCO/PLL1P	CPU-OCD	Up to 120 MHz Division ratio: 1/2/3/4/6/8/12/16/32/64/128/256
SCI clock (SCICLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL1P/ PLL1Q/PLL1R/ PLL2P/PLL2Q/PLL2R	SCI	Up to 120 MHz Division ratio: 1/2/3/4/5/6/8
SPI clock (SPICLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL1P/PLL1Q/PLL1R/ PLL2P/ PLL2Q/PLL2R	SPI	Up to 120 MHz Division ratio: 1/2/3/4/5/6/8
Octal-SPI clock (OCTACLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL1P/PLL1Q/PLL1R/ PLL2P/ PLL2Q/PLL2R	Octal-SPI	Up to 200 MHz Division ratio: 1/2/3/4/5/6/8
Octal-SPI divide clock (OCTADIVCLK)	1/2 OCTACLK	Octal-SPI	Up to 100 MHz
CANFD core clock (CANFDCLK)	MOSC/SOSC/HOCO/MOCO/ LOCO/PLL1P/PLL1Q/PLL1R/ PLL2P/ PLL2Q/PLL2R	CANFD	Up to 80 MHz Division ratio: 1/2/3/4/5/6/8
USB clock (USBCLK)	MOSC/HOCO/MOCO/PLL1P/ PLL1Q/PLL1R/PLL2P/PLL2Q/ PLL2R	USBFS	48 MHz Division ratio: 1/2/3/4/5/6/8
Clock/buzzer output (CLKOUT)	MOSC/SOSC/HOCO/MOCO/ LOCO	CLKOUT pin	Up to 60 MHz Division ratio: 1/2/4/8/16/32/64/128
CAN clock (CANMCLK)	MOSC	CANFD	8 MHz to 48 MHz No division
ULPT LOCO clock (ULPTLCLK)	LOCO	ULPT	32.768 kHz No division



**Table 8.2 Clock generation circuit specifications for the internal clocks (2 of 2)**

Item	Clock source	Clock supply	Specification
ULPT Sub-clock (ULPTSCLK)	SOSC	ULPT	32.768 kHz No division
AGT LOCOclock (AGTLCLK)	LOCO	AGT	32.768 kHz No division
AGT Sub clock (AGTSCLK)	SOSC	AGT	32.768 kHz No division
CAC Main clock (CACMCLK)	MOSC	CAC	Up to 48 MHz No division
CAC Sub clock (CACSCLK)	SOSC	CAC	32.768 kHz No division
CAC HOCO clock (CACHCLK)	HOCO	CAC	16/18/20/32/48 MHz No division
CAC MOCO clock (CACMOCLK)	MOCO	CAC	8 MHz No division
CAC LOCO clock (CACLCLK)	LOCO	CAC	32.768 kHz No division
RTC LOCO clock (RTCLCLK)	LOCO	RTC	32.768 kHz No division
RTC Sub clock (RTCSCLK)	SOCO	RTC	32.768 kHz No division
IWDT clock (IWDTCLK)	1/2 LOCO	IWDT	16.384 kHz No division
SysTick timer clock (SYSTICKCLK)	1/8 MOCO	SysTick timer	1 MHz No division
JTAG clock (JTAGTCK)	TCK	JTAG	Up to 25 MHz No division
Serial wire clock (SWCLK)	TCK	OCD	Up to 25 MHz No division
TCLK pin output (TCLK)	1/2 TRCLK	TCLK pin	Up to 60 MHz

Note: When selecting PLL as the clock source, set the PLL output frequency to 480 MHz or less.

Note: If the PLL reference clock source is HOCO, the PLL multiplication setting must be set within output frequency range of PLL in consideration of HOCO frequency (minimum/maximum).

Note: Restrictions on setting clock frequency: CPUCLK  $\geq$  ICLK, ICLK  $\geq$  PCLKA  $\geq$  PCLKB, ICLK  $\geq$  FCLK, PCLKD  $\geq$  PCLKA  $\geq$  PCLKB  
Restrictions on clock frequency ratio: (N: integer, and up to 64)

CPUCLK:ICLK = N:1, ICLK:FCLK = N:1, ICLK:PCLKA = N:1, ICLK:PCLKB = N:1, ICLK:PCLKC = N:1 or 1:N, ICLK:PCLKD = N:1 or 1:N, ICLK:PCLKE = N:1 or 1:N

When one clock selects 3, 6, 12-divisions, the other clocks set by SCKDIVCR and SCKDIVCR2 registers must not select 2, 4, 8, 16, 32, 64- divisions.

If the A/D converter is enabled, the clock frequency ratio is constrained as follows:

PCLKA:PCLKC = 1:1 or 2:1 or 4:1 or 8:1 or 1:2 or 1:4

If the CAN-FD is used, clock frequency ratio is constrained to be PCLKA:PCLKE = 1:2.

Note: Restrictions on the minimum FCLK frequency 4MHz when P/E.

Note: The multiplication of PLL1 and PLL2 should be set to be within the output frequency range of PLL1 and PLL2, taking the frequency of HOCO into consideration when not using the FLL function.

Note: Clocks have a permissible frequency range (See [Table 8.2](#)).

Flash memory and SRAM also have a permissible operating frequency range in each wait cycle setting. (See [section 47, SRAM](#), [section 49, Flash Memory](#))

Those clock frequency ranges must be satisfied even if the HOCO has its maximum or minimum frequency when not using FLL function. (See [section 52, Electrical Characteristics](#)).

Note: When using ETHERC, the PCLKA frequency is as follows:

12.5 MHz  $\leq$  PCLKA  $\leq$  120 MHz



**Table 8.3 Input/Output Pins of Clock Generation Circuit**

Pin name	I/O	Description
XTAL	Output	These pins are used to connect a ceramic resonator or crystal resonator. The EXTAL pin can also be used to input an external clock. For details, see <a href="#">section 8.3.2. External Clock Input</a> .
EXTAL	Input	
XCIN	Input	These pins are used to connect a 32.768-kHz crystal resonator
XCOU	Output	
EXCIN	Input	External sub-clock input
TCK/SWCLK	Input	This pin is used to input the clock for the JTAG/SWD
CLKOUT	Output	This pin is used to output the CLKOUT clock
TRCLK	Output	TRACE clock output.

## 8.2 Register Descriptions

### 8.2.1 CGFSAR : Clock Generation Function Security Attribute Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x3C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	NONS EC22	NONS EC21	—	—	NONS EC18	NONS EC17	NONS EC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	NONS EC11	—	NONS EC09	NONS EC08	NONS EC07	NONS EC06	NONS EC05	NONS EC04	NONS EC03	NONS EC02	—	NONS EC00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC00	Non Secure Attribute bit 00 Target register: SCKDIVCR, SCKDIVCR2, SCKSCR Target factor: system clock control 0: Secure 1: Non Secure	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	NONSEC02	Non Secure Attribute bit 02 Target register: HOCOCCR, HOCOCCR2, FLLCR1, FLLCR2, HOCOUTCR, HOCOSCR Target factor: HOCO 0: Secure 1: Non Secure	R/W
3	NONSEC03	Non Secure Attribute bit 03 Target register: MOCOCCR, MOCOUTCR Target factor: MOCO 0: Secure 1: Non Secure	R/W
4	NONSEC04	Non Secure Attribute bit 04 Target register: LOCOCCR, LOCOUTCR Target factor: LOCO 0: Secure 1: Non Secure	R/W
5	NONSEC05	Non Secure Attribute bit 05 Target register: MOSCCR, MOSCWTCR, MOMCR, MOSCSR Target factor: MOSC 0: Secure 1: Non Secure	R/W

Bit	Symbol	Function	R/W
6	NONSEC06	Non Secure Attribute bit 06 Target register: OSTDCR, OSTDSR Target factor: oscillation stop detection control 0: Secure 1: Non Secure	R/W
7	NONSEC07	Non Secure Attribute bit 07 Target register: SOSCCR, SOMCR Target factor: SOSC 0: Secure 1: Non Secure	R/W
8	NONSEC08	Non Secure Attribute bit 08 Target register: PLLCCR, PLLCCR2, PLLCR Target factor: PLL 0: Secure 1: Non Secure	R/W
9	NONSEC09	Non Secure Attribute bit 09 Target register: PLL2CCR, PLL2CCR2, PLL2CR Target factor: PLL2 0: Secure 1: Non Secure	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	NONSEC11	Non Secure Attribute bit 11 Target register: CKOCR Target factor: CLKOUT control 0: Secure 1: Non Secure	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	NONSEC16	Non Secure Attribute bit 16 Target register: USBCKDIVCR, USBCKCR Target factor: USBCLK 0: Secure 1: Non Secure	R/W
17	NONSEC17	Non Secure Attribute bit 17 Target register: OCTACKDIVCR, OCTACKCR Target factor: OCTACKL 0: Secure 1: Non Secure	R/W
18	NONSEC18	Non Secure Attribute bit 18 Target register: CANFDCKDIVCR, CANFDCKCR Target factor: CANFDCLK 0: Secure 1: Non Secure	R/W
20:19	—	These bits are read as 0. The write value should be 0.	R/W
21	NONSEC21	Non Secure Attribute bit 21 Target register: SCICKDIVCR, SCICKCR Target factor: SCICLK 0: Secure 1: Non Secure	R/W
22	NONSEC22	Non Secure Attribute bit 22 Target register: SPIICKDIVCR, SPIICKCR Target factor: SPICLK 0: Secure 1: Non Secure	R/W
31:23	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

Note: Set the PRCR.PRC4 bit to 1 (write enabled) before rewriting this register.

CGFSAR register controls the secure attribute of Clock Generation Function registers.

**NONSEC00 bit (Non Secure Attribute bit 00)**

This bit controls the security attribute of SCKDIVCR, SCKDIVCR2, SCKSCR.

**NONSEC02 bit (Non Secure Attribute bit 02)**

This bit controls the security attribute of HOCOCCR, HOCOCCR2, FLLCR1, FLLCR2, HOCOUTCR, HOCOSCR.

**NONSEC03 bit (Non Secure Attribute bit 03)**

This bit controls the security attribute of MOCOCCR, MOCOUTCR.

**NONSEC04 bit (Non Secure Attribute bit 04)**

This bit controls the security attribute of LOCOCCR, LOCOUTCR.

**NONSEC05 bit (Non Secure Attribute bit 05)**

This bit controls the security attribute of MOSCCR, MOSCWTCR, MOMCR, MOSCSR.

**NONSEC06 bit (Non Secure Attribute bit 06)**

This bit controls the security attribute of OSTDCR, OSTDSR.

**NONSEC07 bit (Non Secure Attribute bit 07)**

This bit controls the security attribute of SOSCCR, SOMCR.

**NONSEC08 bit (Non Secure Attribute bit 08)**

This bit controls the security attribute of PLLCCR, PLLCCR2, PLLCR.

**NONSEC09 bit (Non Secure Attribute bit 09)**

This bit controls the security attribute of PLL2CCR, PLL2CCR2, PLL2CR.

**NONSEC11 bit (Non Secure Attribute bit 11)**

This bit controls the security attribute of CKOCCR.

**NONSEC16 bit (Non Secure Attribute bit 16)**

This bit controls the security attribute of USBCKDIVCR, USBCKCR.

**NONSEC17 bit (Non Secure Attribute bit 17)**

This bit controls the security attribute of OCTACKDIVCR, OCTACKCR.

**NONSEC18 bit (Non Secure Attribute bit 18)**

This bit controls the security attribute of CANFDCKDIVCR, CANFDCKCR.

**NONSEC21 bit (Non Secure Attribute bit 21)**

This bit controls the security attribute of SCICKDIVCR, SCICKCR.

**NONSEC22 bit (Non Secure Attribute bit 22)**

This bit controls the security attribute of SPICKDIVCR, SPICKCR.

### 8.2.2 SCKDIVCR : System Clock Division Control Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	FCK[3:0]				ICK[3:0]				PCKE[3:0]				BCK[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PCKA[3:0]				PCKB[3:0]				PCKC[3:0]				PCKD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PCKD[3:0] <sup>*4</sup>	Peripheral Module Clock D (PCLKD) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W
7:4	PCKC[3:0] <sup>*4</sup>	Peripheral Module Clock C (PCLKC) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W
11:8	PCKB[3:0] <sup>*3</sup>	Peripheral Module Clock B (PCLKB) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W

Bit	Symbol	Function	R/W
15:12	PCKA[3:0] <sup>*3</sup>	Peripheral Module Clock A (PCLKA) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W
19:16	BCK[3:0] <sup>*2</sup>	Unnecessary clock (BCLK) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Settings prohibited	R/W
23:20	PCKE[3:0] <sup>*4</sup>	Peripheral Module Clock E (PCKE) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Settings prohibited	R/W
27:24	ICK[3:0] <sup>*1*2*3*4*5</sup>	System Clock (ICK) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W
31:28	FCK[3:0] <sup>*1</sup>	FlashIF Clock (FCLK) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note: When one clock selects 3, 6, 12-divisions, the other clocks set by SCKDIVCR and SCKDIVCR2 registers must not select 2, 4, 8, 16, 32, 64- divisions.

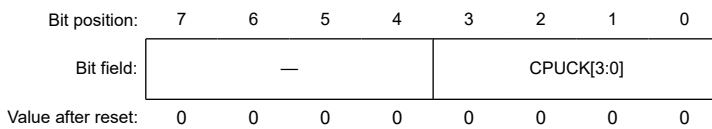
- Note 1. The following relation is required between the frequencies of the system clock (ICLK) and the FlashIF clock (FCLK).  
ICLK:FCLK=N:1 (N: integer)
- Note 2. The following relation is required between the frequencies of the system clock (ICLK) and the unnecessary clock (BCLK).  
When the ICLK frequency is more than 120 MHz, ICLK:BCLK = N:1 (N: integer and N ≥ 2)  
When the ICLK frequency is 120 MHz or less, ICLK:BCLK = N:1 (N: integer)
- Note 3. The following relation is required between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKA, PCLKB)  
ICLK:PCLKA = N:1, ICLK:PCLKB = N:1 (N: integer)
- Note 4. The following relation is required between the frequencies of the system clock (ICLK) and the peripheral module clocks (PCLKC, PCLKD, PCLK E):  
ICLK:PCLKC, PCLKD PCLK E = N:1 or 1:N (N: integer)
- Note 5. The following relation is required between the frequencies of the CPU clock (CPUCLK) and the system clock (ICLK):  
CPUCLK:ICLK = N:1 (N: integer)

SCKDIVCR selects the frequencies of the system clock (ICLK), peripheral module clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLK E), FlashIF clock (FCLK).

### 8.2.3 SCKDIVCR2 : System Clock Division Control Register 2

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x24



Bit	Symbol	Function	R/W
3:0	CPUCK[3:0] <sup>*1</sup>	CPU Clock (CPUCLK) Select 0 0 0 0: × 1/1 0 0 0 1: × 1/2 0 0 1 0: × 1/4 0 0 1 1: × 1/8 0 1 0 0: × 1/16 0 1 0 1: × 1/32 0 1 1 0: × 1/64 1 0 0 0: × 1/3 1 0 0 1: × 1/6 1 0 1 0: × 1/12 Others: Setting prohibited.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: When one clock selects 3, 6, 12-divisions, the other clocks set by SCKDIVCR and SCKDIVCR2 registers must not select 2, 4, 8, 16, 32, 64- divisions.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

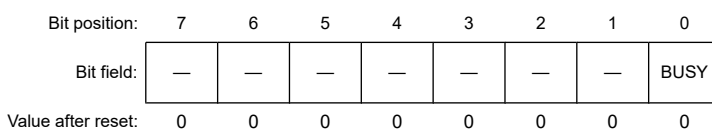
Note 1. The following relation is required between the frequencies of the CPU clock (CPUCLK) and the system clock (ICLK):  
CPUCLK:ICLK = N:1 (N: integer)

SCKDIVCR2 selects the frequencies of the cpu clock (CPUCLK).

### 8.2.4 SYRACCR : System Register Access Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xCC





Bit	Symbol	Function	R/W
0	BUSY	Access Ready monitor 0: Ready to read/write access 1: Writing in progress	R
7:1	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-5, P-TYPE-5

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SYRACCR register monitor that the target registers can be accessed.

The target registers are LOCOCR and LOCOUTCR.

### BUSY bit (Access Ready monitor)

When SYRACCR.BUSY is 0, you can access the target registers.

Make sure that SYRACCR.BUSY is 0 before accessing the target registers.

When SYRACCR.BUSY is 1, the write access to one of the target registers is in progress. If SYRACCR.BUSY is 1, further writing to one of the target registers is ignored, and the read value from the target registers is not guaranteed.

## 8.2.5 SCKSCR : System Clock Source Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x026

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	CKSEL[2:0]	
------------	---	---	---	---	---	------------	--

Value after reset: 0 0 0 0 0 0 0 1

Bit	Symbol	Function	R/W
2:0	CKSEL[2:0]	Clock Source Select 0 0 0: HOCO 0 0 1: MOCO (Value after reset) 0 1 0: Setting prohibited 0 1 1: Main clock oscillator (MOSC) 1 0 0: Sub-clock oscillator (SOSC) 1 0 1: PLL1 output clock P (PLL1P) 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note: S-TYPE-3, P-TYPE-2

The SCKSCR register selects system clock source.

### CKSEL[2:0] bits (Clock Source Select)

The CKSEL[2:0] bits select the source for the following modules:

- CPU clock (CPUCLK)
- System clock (ICLK)
- Peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD and PCLKE)
- FlashIF clock (FCLK)

The bits select from one of the following sources:

- Middle-speed on-chip oscillator (MOCO)
- High-speed on-chip oscillator (HOCO)

- Main clock oscillator (MOSC)
- Sub-clock oscillator (SOSC)
- PLL1 output clock P (PLL1P)

The operating state of each clock source is controlled not only by the clock oscillation enable settings but also by the operating modes of the product. Some clock sources might be forcibly stopped depending on the product operating mode being used.

Check the operation state of clock sources in each product operating mode, and do not select the clock source to be stopped in SCKSCR. The clock sources should be switched when there are no occurring internal asynchronous interrupt. For details, see [section 10, Low Power Modes](#).

### 8.2.6 PLLCCR : PLL Clock Control Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x028



Bit	Symbol	Function	R/W
1:0	PLIDIV[1:0] <sup>*1</sup>	PLL1 Input Frequency Division Ratio Select 0 0: 1/1 0 1: 1/2 1 0: 1/3 1 1: 1/4	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	PLSRCSEL	PLL1 Clock Source Select 0: Main clock oscillator 1: HOCO <sup>*3</sup>	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
7:6	PLLMULNF[1:0] <sup>*2</sup>	PLL1 Frequency Multiplication Fractional Factor Select 0 0: 0.00 (Value after reset) 0 1: 0.33 (1/3) 1 0: 0.66 (2/3) 1 1: 0.50 (1/2)	R/W
15:8	PLLMUL[7:0] <sup>*2</sup>	PLL1 Frequency Multiplication Factor Select 0x19: × 26 (Value after reset) 0x34: × 53 0x35: × 54 ⋮ 0x58: × 89 0x59: × 90 0x5A: × 91 ⋮ 0xB2: × 179 0xB3: × 180 Others: Setting prohibited.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note 1. PLIDIV[1:0] should be set so that the frequency of PLL1 input signal is within the range of [section 8.1. Overview](#).

Note 2. PLLMUL[7:0] and PLLMULNF[1:0] should be set so that the frequency of PLL1 output signal is within the range of [section 8.1. Overview](#).

Note 3. The FLL function must be enabled when using USBCLK.

The PLLCCR register sets the operation of the PLL1 circuit.

Writing to the PLLCCR is prohibited when the PLLCR.PLLSTP bit is 0 (the PLL1 operates).

**PLDIV[1:0] bits (PLL1 Input Frequency Division Ratio Select)**

These bits select the frequency division ratio of the PLL1 clock source.

**PLSRCSEL bit (PLL1 Clock Source Select)**

This bit selects the clock source for the PLL1.

**PLLMULNF[1:0] bit (PLL1 Frequency Multiplication Fractional Factor Select)**

These bits select the fractional part of the frequency multiplication factor for the PLL1 circuit.

**PLLMUL[7:0] bits (PLL1 Frequency Multiplication Factor Select)**

These bits select the frequency multiplication factor of the PLL1 circuit.

**8.2.7 PLLCCR2 : PLL Clock Control Register 2**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x4C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—			PLODIVR[3:0]				PLODIVQ[3:0]				PLODIVP[3:0]				
Value after reset:	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1

Bit	Symbol	Function	R/W
3:0	PLODIVP[3:0] <sup>*1</sup>	PLL1 Output Frequency Division Ratio Select for output clock P 0001: × 1/2 0011: × 1/4 0101: × 1/6 (Value after reset) 0111: × 1/8 1111: × 1/16 Others: Setting prohibited.	R/W
7:4	PLODIVQ[3:0] <sup>*1</sup>	PLL1 Output Frequency Division Ratio Select for output clock Q 0001: × 1/2 0010: × 1/3 0011: × 1/4 0100: × 1/5 0101: × 1/6 (Value after reset) 0111: × 1/8 1000: × 1/9 Others: Setting prohibited.	R/W
11:8	PLODIVR[3:0] <sup>*1</sup>	PLL1 Output Frequency Division Ratio Select for output clock R 0001: × 1/2 0010: × 1/3 0011: × 1/4 0100: × 1/5 0101: × 1/6 (Value after reset) 0111: × 1/8 1000: × 1/9 Others: Setting prohibited.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note 1. It must be set so that the frequency of PLL1 output signal is within the range listed in [Table 8.1](#).

The PLLCCR2 register sets up the operation of the PLL1 circuit. Writing to the PLLCCR2 is prohibited when the PLLCR.PLLSTP bit is 0 (the PLL1 operates).

Even if only one of the PLL1 output clock P, Q, or R is used, PLLCCR2 must be set so that the frequency of PLL1 output signal is within the range listed in [Table 8.1](#).

**PLODIVP[3:0] bit (PLL1 Output Frequency Division Ratio Select for output clock P)**

These bits select the output frequency division ratio for PLL1 output clock P (PLL1P).

**PLODIVQ[3:0] bit (PLL1 Output Frequency Division Ratio Select for output clock Q)**

These bits select the output frequency division ratio for PLL1 output clock Q (PLL1Q).

**PLODIVR[3:0] bit (PLL1 Output Frequency Division Ratio Select for output clock R)**

These bits select the output frequency division ratio for PLL1 output clock R (PLL1R).

**8.2.8 PLLCR : PLL Control Register**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x02A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PLLST P
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PLLSTP	PLL1 Stop Control 0: PLL1 is operating 1: PLL1 is stopped	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The PLLCR register controls the operation of the PLL1 circuit.

**PLLSTP bit (PLL1 Stop Control)**

This bit runs or stops the PLL1 circuit.

If the main clock oscillator is to be selected as the clock source for the PLL1 by the PLLCCR.PLSRCSEL bit, the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set.

After the PLLSTP bit setting is changed to run the PLL1, only use the PLL1 clock after confirming that the OSCSF.PLLSF bit is set to 1. That is, a fixed time for stabilization is required after starting the PLL1 operation. A fixed time is also required for oscillation to stop after stopping the PLL1 operation. Additionally, apply the following limitations when starting and stopping the PLL1 operation by the PLLSTP bit:

- After stopping the PLL1, confirm that the OSCSF.PLLSF bit is 0 before restarting the PLL1.
- Confirm that the PLL1 is operating and that the OSCSF.PLLSF bit is 1 before stopping the PLL1.
- Regardless of whether the PLL1 clock is selected as the system clock source, confirm that the OSCSF.PLLSF is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode after operating the PLL1.
- When transitioning to Software Standby or Deep Software Standby mode after stopping the PLL1, confirm that the OSCSF.PLLSF bit is cleared to 0 before executing a WFI instruction.

Writing 1 to the PLLSTP bit is prohibited when SCKSCR.CKSEL[2:0] = 101 (system clock source = PLL1 output clock P (PLL1P)).

Confirm the following conditions before writing 0 to PLLSTP:

- When PLL1 source clock = MOSC: MOSCCR.MOSTP = 0 (MOSC is enabled)
- When PLL1 source clock = HOCO: HOCOCCR.HCSTP = 0 (HOCO is enabled).

### 8.2.9 PLL2CCR : PLL2 Clock Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x048

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PLL2MUL[7:0]							PLL2MULNF[1:0]	—	PL2SRCSEL	—	—	PL2IDIV[1:0]			
Value after reset:	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PL2IDIV[1:0] <sup>*1</sup>	PLL2 Input Frequency Division Ratio Select 0 0: 1/1 (Value after reset) 0 1: 1/2 1 0: 1/3 1 1: 1/4	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	PL2SRCSEL	PLL2 Clock Source Select 0: Main clock oscillator 1: HOCO <sup>*3</sup>	R/W
5	—	These bits are read as 0. The write value should be 0.	R/W
7:6	PLL2MULNF[1:0] <sup>*2</sup>	PLL2 Frequency Multiplication Fractional Factor Select 00: 0.00 (Value after reset) 01: 0.33 (1/3) 10: 0.66 (2/3) 11: 0.50 (1/2)	R/W
15:8	PLL2MUL[7:0] <sup>*2</sup>	PLL2 Frequency Multiplication Factor Select 0x19: × 26 (Value after reset) 0x34: × 53 0x35: × 54 ⋮ 0x58: × 89 0x59: × 90 0x5A: × 91 ⋮ 0xD2: × 179 0xD3: × 180 Others: Setting prohibited.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note 1. PL2IDIV[1:0] should be set so that the frequency of PLL2 input signal is within the range of [section 8.1. Overview](#).

Note 2. PLL2MUL[7:0] and PLL2MULNF[1:0] should be set so that the frequency of PLL2 output signal is within the range of [section 8.1. Overview](#).

Note 3. The FLL function must be enabled when using USBCLK.

The PLL2CCR register sets the operation of the PLL2 circuit.

Writing to the PLL2CCR register is prohibited when the PLL2CR.PLL2STP bit is 0 (the PLL2 operates).

#### PL2IDIV[1:0] bits (PLL2 Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL2 clock source.

#### PL2SRCSEL bit (PLL2 Clock Source Select)

This bit selects the clock source for the PLL2.

#### PLL2MULNF[1:0] bit (PLL2 Frequency Multiplication Fractional Factor Select)

These bits select the fractional part of the frequency multiplication factor for the PLL2 circuit.

**PLL2MUL[7:0] bits (PLL2 Frequency Multiplication Factor Select)**

These bits select the frequency multiplication factor of the PLL2 circuit.

**8.2.10 PLL2CCR2 : PLL2 Clock Control Register 2**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x4E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—				PL2ODIVR[3:0]				PL2ODIVQ[3:0]				PL2ODIVP[3:0]			
Value after reset:	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0	1

Bit	Symbol	Function	R/W
3:0	PL2ODIVP[3:0] <sup>*1</sup>	PLL2 Output Frequency Division Ratio Select for output clock P 0001: × 1/2 0011: × 1/4 0101: × 1/6 (Value after reset) 0111: × 1/8 1111: × 1/16 Others: Setting prohibited.	R/W
7:4	PL2ODIVQ[3:0] <sup>*1</sup>	PLL2 Output Frequency Division Ratio Select for output clock Q 0001: × 1/2 0010: × 1/3 0011: × 1/4 0100: × 1/5 0101: × 1/6 (Value after reset) 0111: × 1/8 1000: × 1/9 Others: Setting prohibited.	R/W
11:8	PL2ODIVR[3:0] <sup>*1</sup>	PLL2 Output Frequency Division Ratio Select for output clock R 0001: × 1/2 0010: × 1/3 0011: × 1/4 0100: × 1/5 0101: × 1/6 (Value after reset) 0111: × 1/8 1000: × 1/9 Others: Setting prohibited.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note 1. It must be set so that the frequency of PLL2 output signal is within the range listed in [Table 8.1](#).

The PLL2CCR2 register sets up the operation of the PLL2 circuit. Writing to the PLL2CCR2 is prohibited when the PLL2CR.PLL2STP bit is 0 (the PLL2 operates).

Even if only one of the PLL2 output clock P, Q, or R is used, PLL2CCR2 must be set so that the frequency of PLL2 output signal is within the range listed in [Table 8.1](#).

**PL2ODIVP[3:0] bit (PLL2 Output Frequency Division Ratio Select for output clock P)**

These bits select the output frequency division ratio for PLL2 output clock P (PLL2P).

**PL2ODIVQ[3:0] bit (PLL2 Output Frequency Division Ratio Select for output clock Q)**

These bits select the output frequency division ratio for PLL2 output clock Q (PLL2Q).

**PL2ODIVR[3:0] bit (PLL2 Output Frequency Division Ratio Select for output clock R)**

These bits select the output frequency division ratio for PLL2 output clock R (PLL2R).

### 8.2.11 PLL2CR : PLL2 Control Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x04A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PLL2S TP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PLL2STP	PLL2 Stop Control 0: PLL2 is operating 1: PLL2 is stopped	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The PLL2CR register controls the operation of the PLL2 circuit.

#### PLL2STP bit (PLL2 Stop Control)

This bit runs or stops the PLL2 circuit.

If the main clock oscillator is to be selected as the clock source for the PLL2 by the PLL2CCR.PL2SRCSEL bit, the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set.

After the PLL2STP bit setting is changed to run the PLL2, only use the PLL2 clock after confirming that the OSCSF.PLL2SF bit is set to 1. That is, a fixed time for stabilization is required after starting the PLL2 operation. A fixed time is also required for oscillation to stop after stopping the PLL2 operation. Additionally, apply the following limitations when starting and stopping the PLL2 operation by the PLL2STP bit:

- After stopping the PLL2, confirm that the OSCSF.PLL2SF bit is 0 before restarting the PLL2.
- Confirm that the PLL2 is operating and that the OSCSF.PLL2SF bit is 1 before stopping the PLL2.
- Confirm that the OSCSF.PLL2SF bit is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode after operating the PLL2.
- When transitioning to Software Standby or Deep Software Standby mode after stopping the PLL2, confirm that the OSCSF.PLL2SF bit is cleared to 0 before executing a WFI instruction.

Confirm the following conditions before writing 0 to PLL2STP:

- When the PLL2 source clock = MOSC: MOSCCR.MOSTP = 0 (MOSC is enabled)
- When the PLL2 source clock = HOCO: HOCOCCR.HCSTP = 0 (HOCO is enabled).

### 8.2.12 MOSCCR : Main Clock Oscillator Control Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x032

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MOST P
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	MOSTP	Main Clock Oscillator Stop 0: Operate the main clock oscillator*1 1: Stop the main clock oscillator	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note: S-TYPE-3, P-TYPE-2

Note 1. MOMCR register must be set before setting MOSTP to 0.

The MOSCCR register controls the main clock oscillator.

### MOSTP bit (Main Clock Oscillator Stop)

The MOSTP bit starts or stops the main clock oscillator.

When changing the value of the MOSTP bit, execute subsequent instructions only after reading the bit to check that the value is updated.

When using the main clock, the Main Clock Oscillator Mode Oscillation Control Register (MOMCR) and the Main Clock Oscillator Wait Control Register (MOSCWTCR) must be set before setting MOSTP to 0. After setting the MOSTP bit to 0, confirm that the OSCSF.MOSCSF bit is set to 1 before using the main clock oscillator.

A fixed stabilization wait time is required after setting the main clock oscillator to start operation. A fixed wait time is also required for oscillation to stop after stopping the main clock oscillator.

The following restrictions apply when starting and stopping operation:

- After stopping the main clock oscillator, confirm that the OSCSF.MOSCSF bit is 0 before restarting the main clock oscillator
- Confirm that the main clock oscillator operates and that the OSCSF.MOSCSF bit is 1 before stopping the main clock oscillator
- Regardless of whether the main clock oscillator is selected as the system clock source, confirm that the OSCSF.MOSCSF bit is set to 1 before executing a WFI instruction to place the MCU in Software standby after operating the main clock oscillator or Deep Software Standby mode after operating the main clock oscillator.
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting to stop the main clock oscillator, confirm that the OSCSF.MOSCSF bit is set to 0 before executing the WFI instruction.

Writing 1 to MOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC)
- PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P)
- PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and PLLCR.PLLSTP = 0 (PLL1 is operating)
- PLL2CCR.PL2SRCSEL = 0 (PLL2 source clock = MOSC) and PLL2CR.PLL2STP = 0 (PLL2 is operating).

### 8.2.13 SOSCCR : Sub-Clock Oscillator Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xC00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SOSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	SOSTP	Sub-Clock Oscillator Stop 0: Operate the sub-clock oscillator*1 1: Stop the sub-clock oscillator	R/W



Bit	Symbol	Function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note 1. The SOMCR register must be set before setting SOSTP to 0.

The SOSCCR register controls the sub-clock oscillator.

### SOSTP bit (Sub-Clock Oscillator Stop)

The SOSTP bit starts or stops the sub-clock oscillator. When changing the value of the SOSTP bit, only execute subsequent instructions after reading the bit to check that the value is updated. Use the SOSTP bit when using the sub-clock oscillator as the source for a peripheral module, for example the RTC. When using the sub-clock oscillator, set the Sub-Clock Oscillator Mode Control Register (SOMCR) before setting SOSTP to 0.

When selecting External clock input by SOMCR.SOSEL, set SOMCR.SOSEL to 1 and wait 50μs, then set SOSTP to 0.

The following restrictions apply when starting and stopping the operation:

- After stopping the sub-clock oscillator, allow a stop interval of at least 5 SOSC clock cycles before restarting it
- After setting the SOSTP bit to 0, use the sub-clock only after the sub-clock oscillation stabilization time ( $t_{SUBOSCWT}$ ) has elapsed.
- Regardless of whether the sub-clock oscillator is selected as the system clock source, confirm that the sub-clock oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting to stop the sub-clock oscillator, wait for at least 3 SOSC clock cycles before executing the WFI instruction.

Writing 1 to SOSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 100b (system clock source = SOSC).

## 8.2.14 LOCOCR : Low-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x400

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LCSTP	LOCO Stop 0: Operate the LOCO clock 1: Stop the LOCO clock	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note: Make sure that SYRACCR.BUSY is 0 before accessing LOCOCR. When SYRACCR.BUSY is 1, the write access is ignored and the read value is not guaranteed.

The LOCOCR register controls the LOCO clock.

### LCSTP bit (LOCO Stop)

The LCSTP bit starts or stops the LOCO clock.

When changing the value of the bit, only execute subsequent instructions after reading the bit and checking that its value has actually been updated.

## 8.2.15 HOCOCCR : High-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x036

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	HCSTP
Value after reset:	0	0	0	0	0	0	0	0/1 <sup>1</sup>

Bit	Symbol	Function	R/W
0	HCSTP	HOCO Stop 0: Operate the HOCO clock <sup>*2</sup> <sup>3</sup> 1: Stop the HOCO clock	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note 1. The HCSTP bit value after a reset is 0 when the OFS1(\_SEC).HOCOEN bit is 0. It is 1 when the OFS1(\_SEC).HOCOEN bit is 1.

Note 2. If you are using the HOCO (HCSTP = 0), set the OFS1(\_SEC).HOCOFRQ0[2:0] bit to an optimum value.

Note 3. The value of OFS1(\_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCCR2.HCFRQ0[2:0] even if OFS1(\_SEC).HOCOFRQ0[2:0] is not appropriate value.

The HOCOCCR register controls the HOCO clock.

### HCSTP bit (HOCO Stop)

The HCSTP bit starts or stops the HOCO clock.

After setting the HCSTP bit to 0 to start the HOCO clock, confirm that the OSCSF.HOCOSF is set to 1 before using the clock. When OFS1(\_SEC).HOCOEN is set to 0, confirm that OSCSF.HOCOSF is also set to 1 before using the HOCO clock. A fixed stabilization wait time is required after setting the HOCO clock to start operation. A fixed wait time is also required after setting the HOCO clock to stop.

The following limitations apply when starting and stopping operation:

- After stopping the HOCO clock, wait more than HOCO stop width time and confirm that the OSCSF.HOCOSF is 0 before restarting the HOCO clock. See [section 52, Electrical Characteristics](#) for HOCO stop width time.
- Confirm that the HOCO clock operates and that the OSCSF.HOCOSF is 1 before stopping the HOCO clock.
- Regardless of whether the HOCO clock is selected as the system clock source, confirm that the OSCSF.HOCOSF is set to 1 before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode after setting HOCO operation with the HCSTP bit.
- When a transition to Software Standby or Deep Software Standby mode is to follow the setting of the HOCO clock to stop, confirm that the OSCSF.HOCOSF is set to 0 after setting the HOCO clock and before executing the WFI instruction.

Writing 1 to HCSTP is prohibited under the following conditions:

- SCKSCR.CKSEL[2:0] = 000b (system clock source = HOCO)
- PLLCCR.PLSRCSEL = 1 (PLL1 source clock = HOCO) and SCKSCR.CKSEL[2:0] = 101b (system clock source = PLL1P)
- PLLCCR.PLSRCSEL = 1 (PLL1 source clock = HOCO) and PLLCR.PLLSTP = 0 (PLL1 is operating)
- PLL2CCR.PL2SRCSEL = 1 (PLL2 source clock = HOCO) and PLL2CR.PLL2STP = 0 (PLL2 is operating).

## 8.2.16 HOCOCR2 : High-Speed On-Chip Oscillator Control Register2

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x037

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HCFRQ0[2:0]		
Value after reset:	0	0	0	0	0	0/1**	0/1**	0/1**

Bit	Symbol	Function	R/W
2:0	HCFRQ0[2:0]	HOCO Frequency Setting 0 0 0 0: 16MHz 0 0 1: 18MHz 0 1 0: 20MHz 1 0 0: 32MHz 1 1 1: 48MHz Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note 1. Value after reset of the HCFRQ0[2:0] bits depend on OFS1(\_SEC).HOCOFRQ0[2:0] bits.

The HOCOCR2 register controls the HOCO clock.

Writing to the HOCOCR2 is prohibited when the HOCOCR.HCSTP bit is 0 (the HOCO operates)

### HCFRQ0[2:0] bits (HOCO Frequency Setting 0)

These bits select the frequency of HOCO.

## 8.2.17 MOCOCR : Middle-Speed On-Chip Oscillator Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x038

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MCSTP	MOCO Stop 0: MOCO clock is operating 1: MOCO clock is stopped	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The MOCOCR register controls the MOCO clock.

### MCSTP bit (MOCO Stop)

This bit starts or stops the MOCO clock.

When changing the value of the bit, only execute subsequent instructions after reading the bit and checking that its value has actually been updated.

There are restrictions on MOCO operation conditions, MOCO stop conditions and the setting of the MCSTP bit. For details, see [section 8.6. Middle speed On Chip Oscillator \(MOCO\)](#).

## 8.2.18 FLLCR1 : FLL Control Register1

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x039

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FLLCN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLLCN	FLL Enable 0: FLL function is disabled 1: FLL function is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: HOCO must be stopped (HOCOCCR.HCSTP = 1) before FLLCR1.FLLCN is modified.

Note: SOSC must be operating with stabilization while FLL is enabled (FLLCR1.FLLCN = 1).

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The FLLCR1 register controls the FLL function of the HOCO.

### FLLCN bit (FLL Enable)

This bit enables or disables the FLL function of the HOCO.

If FLL is enabled, the frequency accuracy is guaranteed after FLL is stabilized. The FLL stabilization can be checked by the CAC frequency measurement, but it must be executed after HOCO stabilization.

In addition, you must disable FLL by setting the FLLCN bit to 0 before transitioning to Software Standby mode.

Table 8.4 show an example flow of the FLL setting for each case.

**Table 8.4 FLL setting flow (1 of 2)**

Step	Operation
After reset release/ Deep Software Standby cancellation	1 Start (After reset release / Deep Software Standby cancellation)
	2 FLL setting (FLLCR2.FLLCNTL)
	3 Enable FLL (FLLCR1.FLLCN = 1) Note: SOSC must be running with the oscillation stabilization.
	4 Enable HOCO (HOCOCCR.HCSTP = 0)
	5 Wait for the FLL stabilization ( $t_{FLLWT}$ )
	6 Check the HOCO stabilization (OSCSF.HOCOSF = 1)
	7 End (HOCO can be used.)

**Table 8.4 FLL setting flow (2 of 2)**

Step	Operation
Software standby transition/ cancellation	1 Start (FLL is being used.)
	2 Stop HOCO (HOCOCCR.HCSTP = 1) Note: If HOCO is used as the system clock source or the PLL reference clock, these clock source must be changed to another clock before HOCO is stopped.
	3 Disable FLL (FLLCR1.FLLEN = 0)
	4 WFI instruction
	5 Software standby mode
	6 Software standby cancellation
	7 Enable FLL (FLLCR1.FLLEN = 1)
	8 Enable HOCO (HOCOCCR.HCSTP = 0)
	9 Wait for the FLL stabilization ( $t_{FLLWT}$ )
	10 Check the HOCO stabilization (OSCSF.HOCOSF = 1)
	11 End (HOCO can be used.)

### 8.2.19 FLLCR2 : FLL Control Register2

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x03A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	FLLCNTL[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	FLLCNTL[10:0]	FLL Multiplication Control When OFS1(_SEC).HOCOFRQ0[2:0] is 000b (16 MHz) or 100b (32 MHz), these bits must be set to 0x1E9. When OFS1(_SEC).HOCOFRQ0[2:0] is 001b (18 MHz), these bits must be set to 0x226. When OFS1(_SEC).HOCOFRQ0[2:0] is 010b (20 MHz), these bits must be set to 0x263. When OFS1(_SEC).HOCOFRQ0[2:0] is 111b (48 MHz), these bits must be set to 0x1E9. Settings other than above are prohibited.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note: The value of OFS1(\_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCCR2.HCFRQ0[2:0] bits.

The FLLCR2 register controls the FLL function of the HOCO.

#### FLLCNTL[10:0] bits (FLL Multiplication Control)

These bits select the multiplication ratio of the FLL reference clock.

These bits must be set before FLL is enabled (FLLCR1.FLLEN = 1).

## 8.2.20 OSCSF : Oscillation Stabilization Flag Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x03C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PLL2S F	PLLSF	—	MOSC SF	—	—	HOCO SF

Value after reset: 0 0 0 0 0 0 0 0/1<sup>1</sup>

Bit	Symbol	Function	R/W
0	HOCOSF	HOCO Clock Oscillation Stabilization Flag 0: The HOCO clock is stopped or is not yet stable 1: The HOCO clock is stable, so is available for use as the system clock source	R
2:1	—	These bits are read as 0.	R
3	MOSCSF	Main Clock Oscillation Stabilization Flag 0: The main clock oscillator is stopped or is not yet stable <sup>*2</sup> 1: The main clock oscillator is stable, so is available for use as the system clock source	R
4	—	This bit is read as 0.	R
5	PLLSF	PLL1 Clock Oscillation Stabilization Flag 0: The PLL1 clock is stopped or is not yet stable. 1: The PLL1 clock is stable, so is available for use as the system clock source	R
6	PLL2SF	PLL2 Clock Oscillation Stabilization Flag 0: The PLL2 clock is stopped or is not yet stable. 1: The PLL2 clock is stable	R
7	—	This bit is read as 0.	R

Note: S-TYPE-5, P-TYPE-5

Note 1. The value after reset depends on the OFS1(\_SEC).HOCOEN setting.

When OFS1(\_SEC).HOCOEN = 1 (disable HOCO), the value after reset of HOCOSF is 0.

When OFS1(\_SEC).HOCOEN = 0 (enable HOCO), the HOCOSF value is set to 0 immediately after reset is released, and the HOCOSF value is set to 1 after the HOCO oscillation stabilization wait time elapses.

Note 2. This is true when an appropriate value is set in the Wait Control register for the main clock oscillator. If the wait time value is not sufficient, the oscillation stabilization flag is set to 1 and supply of the clock signal to the internal circuits starts before oscillation is stable.

This register is not controlled by CGFSAR register.

The OSCSF register contains flags that indicates the operating status of the oscillation stabilization wait circuit of each oscillator. After oscillation starts, these circuits generate a waiting time until the output clock of each oscillator stabilizes. The status of each oscillation stabilization wait flag set to 1 indicates that the clock supply is stable and can be used in related circuits.

### HOCOSF flag (HOCO Clock Oscillation Stabilization Flag)

The HOCOSF flag indicates the operating status of the counter that measures the wait time for the high-speed clock oscillator (HOCO). When OFS1(\_SEC).HOCOEN is set to 0, confirm that OSCSF.HOCOSF is set to 1 before using the HOCO clock.

[Setting condition]

- After the HOCO clock stops and the HOCOEN.HCSTP bit is set to 0, HOCOSF is set to 1 after HOCO clock is stable to  $\pm 10\%$  or less. If the HOCO clock is used at 1.5%, it must wait for wait time specified in the electrical characteristics. (For details, see [section 52, Electrical Characteristics](#).)  
The HOCO clock is supplied when HOCOEN.HCSTP bit is set to 0, regardless of the state of HOCOSF.

[Clearing condition]

- When the HOCO clock is operating and then is deactivated because the HOCOEN.HCSTP bit is set to 1.

Note: Even if OFS1(\_SEC).HOCOEN has been set to 0, confirm that the OSCSF.HOCOSF has been set to 1 before starting to use the HOCO clock.

Note: In debugging mode, even if HOCO is oscillating, this flag is masked to 0 for the user program debugging when all of following conditions are met simultaneously:

- The Trace Clock is operating enabled by TRCKCR.TRCKEN bit is 1.
- The Trace Clock source is selected HOCO for oscillation in debug mode by TRCKCR.TRCKSEL bit is 1.
- HOCO.CR.HCSTP bit is set to 1.

### MOSCSF flag (Main Clock Oscillation Stabilization Flag)

The MOSCSF flag indicates the operating status of the counter that measures the wait time for the main clock oscillator.

[Setting condition]

- When the main clock oscillator is stopped and the MOSCCR.MOSTP bit is set to 0, and then the number of LOCO clock cycles corresponding to the setting of the MOSCWTCR register is counted and supply of the main clock within the MCU is started.

[Clearing condition]

- When the main clock oscillator is operating and then is deactivated because the MOSCCR.MOSTP bit is set to 1.

### PLLSF flag (PLL1 Clock Oscillation Stabilization Flag)

The PLLSF flag indicates the oscillation stabilization status for the PLL1.

[Setting condition]

- After the PLL1 stops and the PLLCR.PLLSTP bit is set to 0, supply of the PLL1 clock starts after PLL1 clock is stable. If oscillation by the PLL1 clock source selected in the PLLCCR.PLSRCSEL bit is not stable when the PLLSTP bit is set to 0, waiting for the PLL1 oscillation stabilization after the PLL1 clock source oscillation is stabilized.

[Clearing condition]

- When the PLL1 is operating and then is deactivated because the PLLCR.PLLSTP bit is set to 1.

### PLL2SF flag (PLL2 Clock Oscillation Stabilization Flag)

This flag indicates the oscillation stabilization status for the PLL2.

[Setting condition]

- After the PLL2 stops and the PLL2CR.PLL2STP bit is set to 0, supply of the PLL2 clock starts after PLL2 clock is stable. If oscillation by the PLL2 clock source selected in the PLL2CR.PLSRCSEL is not stable when the PLL2STP bit is set to 0, waiting for the PLL2 oscillation stabilization after the oscillation of the PLL2 clock source is stabilized.

[Clearing condition]

- When the PLL2 is operating and then is deactivated because the PLL2CR.PLL2STP bit is set to 1.

## 8.2.21 OSCMONR : Oscillator Monitor Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x43

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	LOCO MON	MOCO MON	—

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
1	MOCOMON	MOCO operation monitor 0: MOCO is set to operate. 1: MOCO is set to stop.	R
2	LOCOMON	LOCO operation monitor 0: LOCO is set to operate. 1: LOCO is set to stop.	R
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-5

This register is not controlled by CGFSAR register.

### MOCOMON bit (MOCO operation monitor)

MOCOMON indicates the operation or stop setting status for MOCO.

[Clearing condition]

- When MOCO operating conditions are met. For details, see [section 8.6.1. MOCO Operating conditions](#).

[Setting condition]

- When MOCO stop conditions are met. For details, see [section 8.6.2. MOCO Stop conditions](#).

### LOCOMON bit (LOCO operation monitor)

LOCOMON indicates the operation or stop setting status for LOCO.

[Clearing condition]

- When LOCO operating conditions are met. For details, see [section 8.5.1. LOCO Operating conditions](#).

[Setting condition]

- When LOCO stop conditions are met. For details, see [section 8.5.2. LOCO Stop conditions](#).

## 8.2.22 OSTDCR : Oscillation Stop Detection Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x040

Bit position:	7	6	5	4	3	2	1	0
Bit field:	OSTD E	—	—	—	—	—	—	OSTDI E

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	OSTDIE	Oscillation Stop Detection Interrupt Enable 0: Disable oscillation stop detection interrupt (do not notify the POEG) 1: Enable oscillation stop detection interrupt (notify the POEG)	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	OSTDE	Oscillation Stop Detection Function Enable 0: Disable oscillation stop detection function 1: Enable oscillation stop detection function	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The OSTDCR register controls the oscillation stop detection function.

### OSTDIE bit (Oscillation Stop Detection Interrupt Enable)

The OSTDIE bit enables the oscillation stop detection function interrupt. It also controls whether oscillation stop detection is reported to the POEG.



If the Oscillation Stop Detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) requires clearing, set the OSTDIE bit to 0 before clearing OSTDF. Wait for at least 2 PCLKB cycles before setting the OSTDIE bit to 1. By reading the I/O register whose access cycle number is defined by PCLKB, it is possible to secure waiting time of 2 or more cycles of PCLKB.

### OSTDE bit (Oscillation Stop Detection Function Enable)

The OSTDE bit enables the oscillation stop detection function.

When the OSTDE bit is 1 (enabled), the MOCO stop bit (MOCOCCR.MCSTP) is set to 0 and the MOCO operation starts. The MOCO clock cannot be stopped while the oscillation stop detection function is enabled. Writing 1 to the MOCOCCR.MCSTP bit (MOCO stopped) is invalid.

When the Oscillation Stop Detection flag in the Oscillation Stop Detection Status Register (OSTDSR.OSTDF) is 1 (main clock oscillation stop detected), writing 0 to the OSTDE bit is invalid.

The following restrictions apply when using the oscillation stop detection function:

In low-speed mode, selecting division by 1, 2, 4, 8 for ICLK, FCLK, PCLKA, PCLKB, PCLKC, PCLKD and PCLKE is prohibited.

### 8.2.23 OSTDSR : Oscillation Stop Detection Status Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x041

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	OSTDF

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	OSTDF	Oscillation Stop Detection Flag 0: Main clock oscillation stop not detected 1: Main clock oscillation stop detected	R/W <sup>1</sup>
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note 1. This bit can only be set to 0. This bit is cleared to 0 by writing 0 after reading it as 1.

The OSTDSR register indicates the stop detection status of the main clock oscillator.

### OSTDF flag (Oscillation Stop Detection Flag)

The OSTDF flag indicates the main clock oscillator status. When this flag is 1, it indicates that the main clock oscillation stop was detected. After this stop is detected, the OSTDF flag is not set to 0 even when the main clock oscillation is restarted. The OSTDF bit is cleared to 0 by writing 0 after reading it as 1.

At least 3 ICLK cycles of wait time are required between writing 0 to OSTDF and reading it as 0. If the OSTDF flag is set to 0 when the main clock oscillation is stopped, the OSTDF flag becomes 0 then returns to 1.

The OSTDF flag cannot be set to 0 under the following conditions:

- SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC)
- PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (System clock source = PLL1P).

The OSTDF flag must be set to 0 after switching the clock source to sources other than the main clock oscillator and PLL1P.

[Setting condition]

- The main clock oscillator is stopped when OSTDCR.OSTDE = 1 (oscillation stop detection function enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKSCR.CKSEL[2:0] bits are neither 011b (system clock source is MOSC) nor, 101b (system clock source is PLL1P) and PLLCCR.PLSRCSEL bit is not 0 (PLL1 source clock is MOSC).

### 8.2.24 MOSCWTCR : Main Clock Oscillator Wait Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x0A2

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MSTS[3:0]			
Value after reset:	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
3:0	MSTS[3:0]	Main Clock Oscillator Wait Time Setting 0x0: Wait time = 3 cycles (11.4 $\mu$ s) 0x1: Wait time = 35 cycles (133.5 $\mu$ s) 0x2: Wait time = 67 cycles (255.6 $\mu$ s) 0x3: Wait time = 131 cycles (499.7 $\mu$ s) 0x4: Wait time = 259 cycles (988.0 $\mu$ s) 0x5: Wait time = 547 cycles (2086.6 $\mu$ s) 0x6: Wait time = 1059 cycles (4039.8 $\mu$ s) 0x7: Wait time = 2147 cycles (8190.2 $\mu$ s) 0x8: Wait time = 4291 cycles (16368.9 $\mu$ s) 0x9: Wait time = 8163 cycles (31139.4 $\mu$ s) Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

#### MSTS[3:0] bits (Main Clock Oscillator Wait Time Setting)

The MSTS[3:0] bits specify the oscillation stabilization wait time for the main clock oscillator.

Set the main clock oscillation stabilization time to a period longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is input externally, set these bits to 0x0 because the oscillation stabilization time is not required.

The wait time set in these bits is counted using: 1 cycle ( $\mu$ s) =  $1/(f_{\text{LOCO}}[\text{MHz}] \times 8) = 1/(0.032768 \times 8) = 3.81$  ( $\mu$ s) (min.). The LOCO clock automatically oscillates when necessary, regardless of the value of the LOCO.LCSTP bit. After the specified wait time elapses, supply of the main clock starts internally in the MCU, and the OSCSF.MOSCSF flag is set to 1. If the specified wait time is short, supply of the main clock starts before oscillation of the clock becomes stable.

Only rewrite the MOSCWTCR register when the MOSCCR.MOSTP bit is 1 and the OSCSF.MOSCSF flag is 0. Do not rewrite this register under any other conditions.

### 8.2.25 MOMCR : Main Clock Oscillator Mode Oscillation Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA50

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MOSE L	—	—	MODRV0[2:0]		—	—
Value after reset:	0	0	0	1	1	0	1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
3:1	MODRV0[2:0]	Main Clock Oscillator Drive Capability 0 Switching 0 0 0: 8 MHz 0 1 1: 8 MHz to 24 MHz 1 0 1: 8 MHz to 48 MHz Others: Setting prohibited	R/W
4	—	This bit is read as 1. The write value should be 1.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	MOSEL	Main Clock Oscillator Switching 0: Resonator 1: External clock input	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: The EXTAL/XTAL pins are also used as ports. In the initial state, the pin is set as a port.

Note: The MOSCCR.MOSTP bit must be 1 (MOSC is stopped) before changing this register.

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

### MODRV0[2:0] bits (Main Clock Oscillator Drive Capability 0 Switching)

The MODRV0[2:0] bits switches the drive capability for the main clock oscillator. The drive capability of Main Clock Oscillator needs to choose suitable drive capability according to the frequency.

### MOSEL bit (Main Clock Oscillator Switching)

The MOSEL bit switches the source for the main clock oscillator.

## 8.2.26 SOMCR : Sub-Clock Oscillator Mode Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xC01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SOSEL	—	—	—	—	SODRV[1:0]	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SODRV[1:0]	Sub-Clock Oscillator Drive Capability Switching 0 0: Standard (12.5pf) 0 1: Lowpower mode 1 (9pf) 1 0: Lowpower mode 2 (7pf) 1 1: Lowpower mode 3 (4pf)	R/W
5:2	—	These bits are read as 0. The write value should be 0.	R/W
6	SOSEL	Sub-Clock Oscillator Switching 0: Resonator 1: External clock input	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The SOMCR register must be modified when SOSCCR.SOSTP is 1 (SOSC is stopped).

### SODRV[1:0] bits (Sub-Clock Oscillator Drive Capability Switching)

The SODRV[1:0] bits switch the drive capability of the sub-clock oscillator.

### SOSEL bit (Sub-Clock Oscillator Switching)

The SOSEL bits switches the source for the Sub-clock oscillator.

For details, see [section 8.4. Sub-Clock Oscillator](#).

If VCC voltage of the MCU is lower than 1.8 V and this bit is set to 1, RTC registers and VBTBKR[n](n = 0 to 127) (VBATT Backup Register) can be accessed 20  $\mu$ s after the setting of this bit.

### 8.2.27 CKOCR : Clock Out Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x03E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CKOEN		CKODIV[2:0]		—	CKOSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
2:0	CKOSEL[2:0]	Clock Out Source Select 0 0 0: HOCO 0 0 1: MOCO (Value after reset) 0 1 0: LOCO 0 1 1: MOSC 1 0 0: SOSC Others: Setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CKODIV[2:0]	Clock Output Frequency Division Ratio 0 0 0: $\times 1/1$ 0 0 1: $\times 1/2$ 0 1 0: $\times 1/4$ 0 1 1: $\times 1/8$ 1 0 0: $\times 1/16$ 1 0 1: $\times 1/32$ 1 1 0: $\times 1/64$ 1 1 1: $\times 1/128$	R/W
7	CKOEN	Clock Out Enable 0: Disable clock out 1: Enable clock out	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

#### CKOSEL[2:0] bits (Clock Out Source Select)

The CKOSEL[2:0] bits select the source of the clock to be output from the CLKOUT pin. When changing the clock source, set the CKOEN bit to 0.

#### CKODIV[2:0] bits (Clock Output Frequency Division Ratio)

The CKODIV[2:0] bits specify the clock division ratio. Set the CKOEN bit to 0 when changing the division ratio.

#### CKOEN bit (Clock Out Enable)

The CKOEN bit enables output from the CLKOUT pin.

When this bit is set to 1, the selected clock is output. When this bit is set to 0, low is output. When changing this bit, confirm that the clock out source clock selected in the CKOSEL[2:0] bits is stable. Otherwise, a glitch might be generated in the output.

Clear this bit before entering Software Standby or Deep Software Standby mode if the selecting clock out source clock is stopped in that mode.

### 8.2.28 LOCOUTCR : LOCO User Trimming Control Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x402

Bit position: 7 6 5 4 3 2 1 0

Bit field: LOCOUTRM[7:0]

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	LOCOUTRM[7:0]	LOCO User Trimming 0x80: -128 0x81: -127 ⋮ 0xFF: -1 0x00: Center Code 0x01: +1 ⋮ 0x7E: +126 0x7F: +127	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Note: Make sure that SYRACCR.BUSY is 0 before accessing LOCOUTCR. When SYRACCR.BUSY is 1, the write access is ignored and the read value is not guaranteed.

Note: MCU operation is not guaranteed when LOCOUTCR is set to a value that causes the LOCO frequency to be outside of the specification range.

Note: After changing LOCOUTCR, it takes certain time until the frequency to be stable. The time is same as the oscillation stabilization time in startup.

The LOCOUTCR register is added to the original LOCO trimming data.

When the ratio of the LOCO frequency and the other oscillation frequency is an integer value, changing the LOCOUTCR value is prohibited.

Changing LOCOUTCR during RTC operation is prohibited.

### 8.2.29 MOCOUTCR : MOCO User Trimming Control Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x061

Bit position: 7 6 5 4 3 2 1 0

Bit field: MOCOUTRM[7:0]

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	MOCOUTRM[7:0]	MOCO User Trimming 0x80: -128 0x81: -127 ⋮ 0xFF: -1 0x00: Center Code 0x01: +1 ⋮ 0x7E: +126 0x7F: +127	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

- Note: MCU operation is not guaranteed when MOCOUTCR is set to a value that causes the MOCO frequency to be outside of the specification range.
- Note: After changing MOCOUTCR, it takes certain time until the frequency to be stable. The time is same as the oscillation stabilization time in startup.

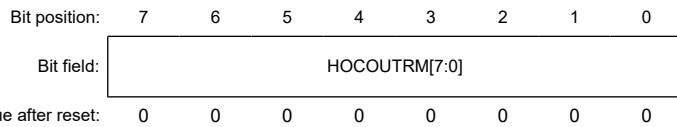
The MOCOUTCR register is added to the original MOCO trimming data.

When the ratio of the MOCO frequency and the other oscillation frequency is an integer value, changing the MOCOUTCR value is prohibited.

### 8.2.30 HOCOUTCR : HOCO User Trimming Control Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x062



Bit	Symbol	Function	R/W
7:0	HOCOUTRM[7:0]	HOCO User Trimming 0x80: -128 0x81: -127 ⋮ 0xFF: -1 0x00: Center Code 0x01: +1 ⋮ 0x7E: +126 0x7F: +127	R/W

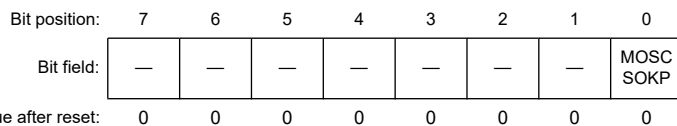
- Note: S-TYPE-3, P-TYPE-2
- Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
- Note: MCU operation is not guaranteed when HOCOUTCR is set to a value that causes the HOCO frequency to be outside of the specification range.
- Note: After changing HOCOUTCR, it takes certain time until the frequency to be stable. The time is same as the oscillation stabilization time in startup.
- Note: These bits must be 0x00 when FLL is enabled (FLLCR1.FLLEN = 1).

The HOCOUTCR register is added to the original HOCO trimming data.

### 8.2.31 MOSCSCR : Main Clock Oscillator Standby Control Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x07C



Bit	Symbol	Function	R/W
0	MOSCSOKP	Main Clock Oscillator Standby Oscillation Keep select 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

- Note: S-TYPE-3, P-TYPE-2
  - Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
- The MOSCSCR must be changed while MOSCCR.MOSTP bit is 1 (MOSC is stopped).

**MOSCSOKP bit (Main Clock Oscillator Standby Oscillation Keep select)**

This bit controls that Main Clock Oscillator keeps oscillating in Software Standby mode.

If this bit is disabled and MOSCCR.MOSTP bit is 0 (MOSC is run), the Main Clock Oscillator stops in Software Standby mode.

If this bit is enabled and MOSCCR.MOSTP bit is 0 (MOSC is run), the Main Clock Oscillator will keep oscillating in Software Standby mode.

If MOSCCR.MOSTP bit is 1 (MOSC is stopped), the Main Clock Oscillator stops in Software Standby mode regardless of the setting of this bit.

**8.2.32 HOCOSCR : High-Speed On-Chip Oscillator Standby Control Register**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x07D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	HOCO SOKP

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	HOCOSOKP	HOCO Standby Oscillation Keep select. 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

**HOCOSOKP bit (HOCO Standby Oscillation Keep select.)**

This bit controls that HOCO keeps oscillating in Software Standby mode.

If this bit is disabled and HOCOCCR.HCSTP bit is 0 (HOCO is run), the HOCO stops in Software Standby mode.

If this bit is enabled and HOCOCCR.HCSTP bit is 0 (HOCO is run), the HOCO will keep oscillating in Software Standby mode.

If HOCOCCR.HCSTP bit is 1 (HOCO is stopped), the HOCO stops in Software Standby mode regardless of the setting of this bit.

**8.2.33 USBCKDIVCR : USB Clock Division Control Register**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x06C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	USBCKDIV[2:0]		

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	USBCKDIV[2:0]	USB Clock (USBCLK) Division Select 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The USBCKDIVCR register controls the USB clock.

### USBCKDIV[2:0] bits (USB Clock (USBCLK) Division Select)

These bits select the frequency of the USB clock (USBCLK) and must be modified when USBCKCR.USBCKSRDY = 1.

## 8.2.34 OCTACKDIVCR : Octal-SPI Clock Division Control Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x06D

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	OCTACKDIV[2:0]	
------------	---	---	---	---	---	----------------	--

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	OCTACKDIV[2:0]	Octal-SPI Clock (OCTACLK) Division Select 0 0 0: 1/1 (Value after reset) 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

OCTACKDIVCR controls the Octal-SPI clock.

### OCTACKDIV[2:0] bits (Octal-SPI Clock (OCTACLK) Division Select)

These bits select the frequency of the Octal-SPI clock (OCTACLK) and must be modified when OCTACKCR.OCTACKSRDY = 1.



### 8.2.35 CANFDCKDIVCR : CANFD Core Clock Division Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x06E

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	CANFDCKDIV[2:0]	
------------	---	---	---	---	---	-----------------	--

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	CANFDCKDIV[2:0]	CANFD core clock (CANFDCLK) Division Select 0 0 0: 1/1 (Value after reset) 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

CANFDCKDIVCR controls the CANFD core clock.

#### CANFDCKDIV[2:0] bits (CANFD core clock (CANFDCLK) Division Select)

These bits select the frequency of the CANFD core clock (CANFDCLK) and must be modified when CANFDCKCR.CANFDCKSRDY = 1.

### 8.2.36 USBCKCR : USB Clock Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x074

Bit position: 7 6 5 4 3 2 1 0

Bit field:	USBC KSRD Y	USBC KSRE Q	—	—	USBCKSEL[3:0]			0
------------	-------------------	-------------------	---	---	---------------	--	--	---

Value after reset: 0 0 0 0 0 0 0 1

Bit	Symbol	Function	R/W
3:0	USBCKSEL[3:0]	USB Clock (USBCLK) Source Select 0000: HOCO 0001: MOCO (Value after reset) 0011: Main clock oscillator 0101: PLL1P 0110: PLL2P 0111: PLL1Q 1000: PLL1R 1001: PLL2Q 1010: PLL2R Others: Setting prohibited.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	USBCKSREQ	USB Clock (USBCLK) Switching Request 0: No request 1: Request switching.	R/W

Bit	Symbol	Function	R/W
7	USBCKSRDY	USB Clock (USBCLK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The USBCKCR register controls the USB clock.

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output. To change the set value of USBCKDIVCR.USBCKDIV[2:0] and USBCKSEL[3:0], use the following procedure:

1. Write 1 to USBCKSREQ.
2. Poll until USBCKSRDY is read as 1. While USBCKSRDY = 1, no clock is output to USBCLK.
3. Write to USBCKDIVCR.USBCKDIV[2:0] and USBCKSEL[3:0].
4. Write 0 to USBCKSREQ.
5. Poll until USBCKSRDY is read as 0.
6. When USBCKSRDY becomes 0, USBCLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when USBCKSREQ = 1 and USBCKSRDY = 0, or when USBCKSREQ = 0 and USBCKSRDY = 1.

#### USBCKSEL[3:0] bits (USB Clock (USBCLK) Source Select)

These bits select the clock source of the USB clock (USBCLK) and must be modified when USBCKCR.USBCKSRDY = 1. Do not stop the oscillator selected by these bits except when MOCO is selected.

#### USBCKSREQ bit (USB Clock (USBCLK) Switching Request)

This bit selects the USBCLK switching request.

#### USBCKSRDY flag (USB Clock (USBCLK) Switching Ready state flag)

This flag indicates the state of switching ready for the USBCLK. When USBCKSRDY = 1, no clock is output to USBCLK.

### 8.2.37 OCTACKCR : Octal-SPI Clock Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x075

Bit position: 7 6 5 4 3 2 1 0

Bit field:	OCTACKSRDY	OCTACKSREQ	—	—	OCTACKSEL[3:0]		
------------	------------	------------	---	---	----------------	--	--

Value after reset: 0 0 0 0 0 0 0 1

Bit	Symbol	Function	R/W
3:0	OCTACKSEL[3:0]	Octal-SPI Clock (OCTACLK) Source Select 0 0 0 0: HOCO 0 0 0 1: MOCO (Value after reset) 0 0 1 0: LOCO 0 0 1 1: Main clock oscillator 0 1 0 0: Sub-clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OCTACKSREQ	Octal-SPI Clock (OCTACLK) Switching Request 0: No request 1: Request switching.	R/W
7	OCTACKSRDY	Octal-SPI Clock (OCTACLK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The OCTACKCR register controls the Octal-SPI clock.

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output.

To change the set value of OCTACKDIVCR.OCTACKDIV[2:0] and OCTACKSEL[3:0], use the following procedure:

1. Write 1 to OCTACKSREQ.
2. Poll until OCTACKSRDY is read as 1. While OCTACKSRDY = 1, no clock is output to OCTACLK.
3. Write to OCTACKDIVCR.OCTACKDIV[2:0] and OCTACKSEL[3:0].
4. Write 0 to OCTACKSREQ.
5. Poll until OCTACKSRDY is read as 0.
6. When OCTACKSRDY becomes 0, OCTACLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when OCTACKSREQ = 1 and OCTACKSRDY = 0, or when OCTACKSREQ = 0 and OCTACKSRDY = 1.

#### **OCTACKSEL[3:0] bits (Octal-SPI Clock (OCTACLK) Source Select)**

These bits select the clock source of the Octal-SPI clock (OCTACLK) and must be modified when OCTACKCR.OCTACKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

#### **OCTACKSREQ bit (Octal-SPI Clock (OCTACLK) Switching Request)**

This bit selects the OCTACLK switching request.

#### **OCTACKSRDY flag (Octal-SPI Clock (OCTACLK) Switching Ready state flag)**

This flag indicates the state of switching ready for the OCTACLK. When OCTACKSRDY = 1, no clock is output to OCTACLK.

### 8.2.38 CANFDCKCR : CANFD Core Clock Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x076

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CANFDCKSRDY	CANFDCKSREQ	—	—	CANFDCKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	CANFDCKSEL[3:0]	CANFD Core Clock (CANFDCLK) Source Select 0 0 0 0: HOCO 0 0 0 1: MOCO (Value after reset) 0 0 1 0: LOCO 0 0 1 1: Main clock oscillator 0 1 0 0: Sub-clock oscillator 0 1 0 1: PLL1P 0 1 1 0: PLL2P 0 1 1 1: PLL1Q 1 0 0 0: PLL1R 1 0 0 1: PLL2Q 1 0 1 0: PLL2R Others: Setting prohibited.	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	CANFDCKSREQ	CANFD Core Clock (CANFDCLK) Switching Request 0: No request 1: Request switching	R/W
7	CANFDCKSRDY	CANFD Core Clock (CANFDCLK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

The CANFDCKCR register controls the CANFD Core Clock (CANFDCLK).

When switching the clock source, ensure that the clock before the switch and the clock after the switch generate stable output. To change the set value of CANFDCKDIVCR.CANFDCKDIV[2:0] and CANFDCKSEL[3:0], use the following procedure:

1. Write 1 to CANFDCKSREQ.
2. Poll until CANFDCKSRDY is read as 1. While CANFDCKSRDY = 1, no clock is output to CANFDCLK.
3. Write to CANFDCKDIVCR.CANFDCKDIV[2:0] and CANFDCKSEL[3:0].
4. Write 0 to CANFDCKSREQ.
5. Poll until CANFDCKSRDY is read as 0.
6. When CANFDCKSRDY becomes 0, CANFDCLK starts to output. Clock switching is complete.

When transitioning to Software Standby or Deep Software Standby mode, do not execute the WFI instruction while performing clock switching. That is, do not execute the WFI instruction when CANFDCKSREQ = 1 and CANFDCKSRDY = 0, or when CANFDCKSREQ = 0 and CANFDCKSRDY = 1.

#### CANFDCKSEL[3:0] bits (CANFD Core Clock (CANFDCLK) Source Select)

These bits select the clock source of the CANFD Core clock (CANFDCLK) and must be modified when CANFDCKCR.CANFDCKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

**CANFDCKSREQ bit (CANFD Core Clock (CANFDCLK) Switching Request)**

This bit selects the CANFDCLK switching request.

**CANFDCKSRDY flag (CANFD Core Clock (CANFDCLK) Switching Ready state flag)**

This flag indicates the state of switching ready for the CANFDCLK. When CANFDCKSRDY = 1, no clock is output to CANFDCLK.

**8.2.39 SCICKDIVCR : SCI clock Division control register**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x054

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SCICKDIV[2:0]		

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	SCICKDIV[2:0]	SCI clock (SCICLK) Division Select 0 0 0: 1/1 (Value after reset) 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SCICKDIVCR controls the SCI clock.

**SCICKDIV[2:0] bits (SCI clock (SCICLK) Division Select)**

These bits select the frequency of the SCI clock (SCICLK) and must be modified when SCICKCR.CKSRDY = 1.

**8.2.40 SPICKDIVCR : SPI clock Division control register**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x056

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SPICKDIV[2:0]		

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	SPICKDIV[2:0]	SPI clock (SPICLK) Division Select 0 0 0: 1/1 (Value after reset) 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/6 1 0 0: 1/8 1 0 1: 1/3 1 1 0: 1/5 Others: Setting prohibited.	R/W

Bit	Symbol	Function	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

SPICKDIVCR controls the SPI clock.

### SPICKDIV[2:0] bits (SPI clock (SPICLK) Division Select)

These bits select the frequency of the SPI clock (SPICLK) and must be modified when SPICKCR.CKSRDY = 1.

#### 8.2.41 SCICKCR : SCI clock control register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x055

Bit position: 7 6 5 4 3 2 1 0

Bit field:	7	6	5	4	3	2	1	0
SCICK SRDY	SCICK SREQ	—	—	SCICKSEL[3:0]				

Value after reset: 0 0 0 0 0 0 0 1

Bit	Symbol	Function	R/W
3:0	SCICKSEL[3:0]0	SCI clock (SCICLK) Source Select 0000: HOCO 0001: MOCO (Value after reset) 0010: LOCO 0011: Main clock oscillator 0100: Sub-clock oscillator 0101: PLL1P 0110: PLL2P 0111: PLL1Q 1000: PLL1R 1001: PLL2Q 1010: PLL2R Others: Setting prohibited.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	SCICKSREQ	SCI clock (SCICLK) Switching Request 0: No request 1: Request switching	R/W
7	SCICKSRDY	SCI clock (SCICLK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: S-TYPE-3, P-TYPE-2

SCICKCR controls the SCI clock.

To change the set value of SCICKDIVCR.SCICKDIV[2:0] and SCICKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure :

1. Write 1 to SCICKSREQ.
2. Polling until SCICKSRDY is read as 1. While SCICKSRDY = 1, no clock is output to SCICLK.
3. Write the setting value to SCICKDIVCR.SCICKDIV[2:0] and SCICKSEL[3:0].
4. Write 0 to SCICKSREQ.
5. Polling until SCICKSRDY is read as 0.
6. When SCICKSRDY becomes 0, SCICLK starts outputting. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with SCICKSREQ = 1 and SCICKSRDY = 0, or SCICKSREQ = 0 and SCICKSRDY = 1.

### SCICKSEL[3:0] bits (SCI clock (SCICLK) Source Select)

These bits select the clock source of the SCI clock (SCICLK) and must be modified when SCICKCR.SCICKSRDY = 1.

Do not stop the oscillator selected by these bits except when MOCO is selected.

### SCICKSREQ bit (SCI clock (SCICLK) Switching Request)

This bit selects the SCICLK switching request.

### SCICKSRDY bit (SCI clock (SCICLK) Switching Ready state flag)

This flag indicates the state of switching ready for the SCICLK.

While SCICKSRDY = 1, no clock is output to SCICLK.

## 8.2.42 SPICKCR : SPI clock control register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x057

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPICKSRDY	SPICKSREQ	—	—	SPICKSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	SPICKSEL[3:0]	SPI clock (SPICLK) Source Select 0000: HOCO 0001: MOCO (Value after reset) 0010: LOCO 0011: Main clock oscillator 0100: Sub-clock oscillator 0101: PLL1P 0110: PLL2P 0111: PLL1Q 1000: PLL1R 1001: PLL2Q 1010: PLL2R Others: Setting prohibited.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	SPICKSREQ	SPI clock (SPICLK) Switching Request 0: No request 1: Request switching	R/W
7	SPICKSRDY	SPI clock (SPICLK) Switching Ready state flag 0: Impossible to Switch 1: Possible to Switch	R

Note: S-TYPE-3, P-TYPE-2

SPICKCR controls the SPI clock.

To change the set value of SPICKDIVCR.SPICKDIV[2:0] and SPICKSEL[3:0], follow the procedure below.

When switching the clock source, it is necessary that the clock before switching and the clock after switching are stably output.

Clock selection switching procedure :

1. Write 1 to SPICKSREQ.
2. Polling until SPICKSRDY is read as 1. While SPICKSRDY = 1, no clock is output to SPICLK.

3. Write the setting value to SPICKDIVCR.SPICKDIV[2:0] and SPICKSEL[3:0].
4. Write 0 to SPICKSREQ.
5. Polling until SPICKSRDY is read as 0.
6. When SPICKSRDY becomes 0, SPICLK starts outputting. Clock switching complete.

When a transition to Software Standby or Deep Software Standby, do not execute the WFI instruction while performing clock selection switching. In other words, do not execute the WFI instruction with SPICKSREQ = 1 and SPICKSRDY = 0, or SPICKSREQ = 0 and SPICKSRDY = 1.

#### SPICKSEL[3:0] bits (SPI clock (SPICLK) Source Select)

These bits select the clock source of the SPI clock (SPICLK) and must be modified when SPICKCR.SPICKSRDY = 1. Do not stop the oscillator selected by these bits except when MOCO is selected.

#### SPICKSREQ bit (SPI clock (SPICLK) Switching Request)

This bit selects the SPICLK switching request.

#### SPICKSRDY bit (SPI clock (SPICLK) Switching Ready state flag)

This flag indicates the state of switching ready for the SPICLK.

While SPICKSRDY = 1, no clock is output to SPICLK.

### 8.2.43 TRCKCR : Trace Clock Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x03F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TRCK EN	—	—	TRCK SEL	TRCK[3:0]			
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
3:0	TRCK[3:0]	Trace Clock operating frequency select 0000: 1/1 0001: 1/2 0010: 1/4 (Value after reset) 0011: 1/8 0100: 1/16 0101: 1/32 0110: 1/64 0111: 1/128 1000: 1/256 1001: 1/3 1010: 1/6 1011: 1/12 Others: Setting prohibited	R/W
4	TRCKSEL	Trace Clock source select 0: System clock source (Value after reset) 1: HOCO (oscillation in debug mode)	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	TRCKEN	Trace Clock operating Enable 0: Stop 1: Operation enable	R/W

Note: S-TYPE5, P-TYPE-2

Trace Clock Control Register controls switching the trace clock.

This register is not controlled by CGFSAR register.



When changing the frequency of the Trace Clock, the Trace Clock must be stopped (TRCKCR.TRCKEN = 0).

#### TRCK[3:0] bit (Trace Clock operating frequency select)

These bits select the frequency of the Trace Clock.

This bit must change when TRCKCR.TRCKEN = 0.

#### TRCKSEL bit (Trace Clock source select)

This bit selects the clock source of the Trace Clock. According to this bit, the Trace Clock is as follows:

[The Case of this bit is set 0]

- The system clock source is selected as the clock source for the trace clock.  
The system clock source is selected by SCKSCR.CKSEL[2:0] bits.
- The clock supply to the trace clock is selected of MOCO divided by 1 when in Software Standby mode or Deep Software Standby mode.

[The Case of this bit is set 1]

- The HOCO is selected as the clock source for the trace clock
- If TRCKSEL is set to 1, HOCO is operated no depending on the state of HOCOCR.HCSTP. However, OSCSF does not confirm that the swing is stable, so wait for the user to wait for the stable wait time
- HOCO continues to oscillate and supply the Trace Clock even if in Software Standby mode or Deep Software Standby mode.

This bit must change when TRCKCR.TRCKEN = 0.

#### TRCKEN bit (Trace Clock operating Enable)

This bit enables or disables the clock supply of the Trace Clock.

TRCK[3:0] bits and TRCKSEL bit must be set before changing this bit from 0 (stop) to 1 (enable).

When changing this bit from 1 (enable) to 0 (stop), Trace function must be disable.

The trace clock is output with this bit set to 1 and under the following conditions.

$CDBGPWRUPREQ = 1 \ \& \ AL \neq AL0 \mid DBGAUTH0.DBGEN0 = 1 \ \& \ OFS1\_SEC.SWDBG = 0.$

### 8.3 Main Clock Oscillator

To supply the clock signal to the main clock oscillator, use one of the following ways:

- Connect an oscillator
- Connect the input of an external clock signal.

#### 8.3.1 Connecting a Crystal Resonator

Figure 8.2 shows an example of connecting a crystal resonator. A damping resistor (Rd) can be added, if required.

Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the manufacturer recommends using an external feedback resistor (Rf), insert an Rf between EXTAL and XTAL by following the instructions.

When connecting a resonator to supply the clock, the frequency of the resonator must be in the frequency range of the resonator for the main clock oscillator as described in Table 8.1.

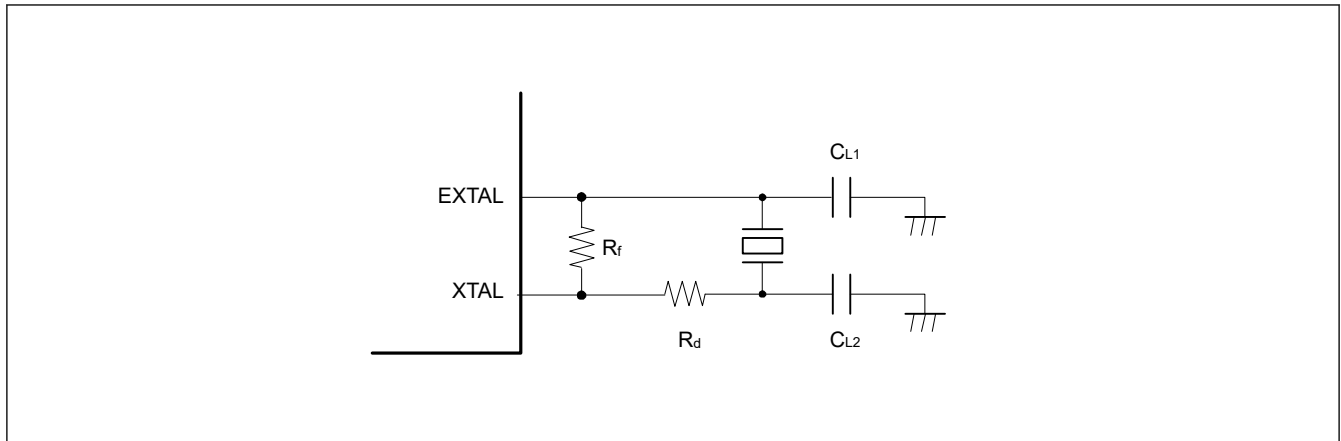


Figure 8.2 Example of crystal resonator connection

Figure 8.3 shows an equivalent circuit of the crystal resonator.

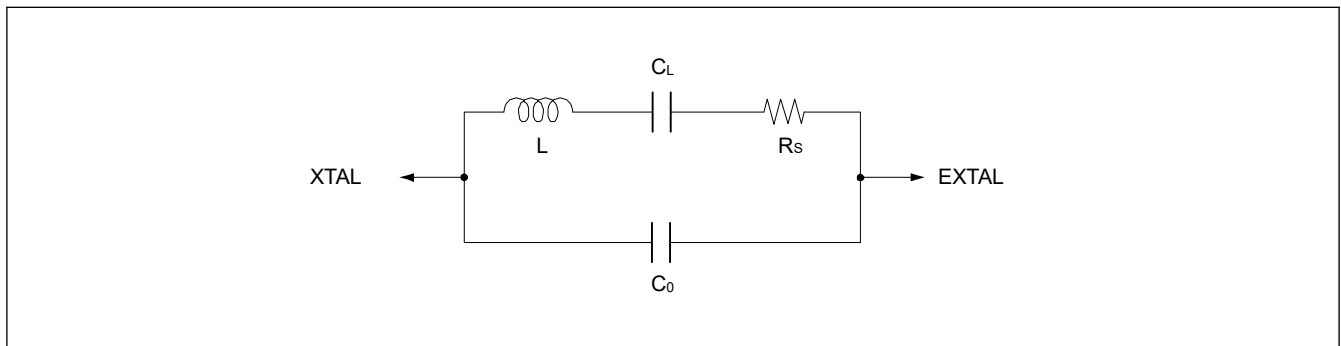


Figure 8.3 Equivalent circuit of the crystal resonator

### 8.3.2 External Clock Input

Figure 8.4 shows an example of connecting an external clock input. To operate the oscillator with an external clock signal, set the MOMCR.MOSEL bit to 1. The XTAL pin becomes high impedance.

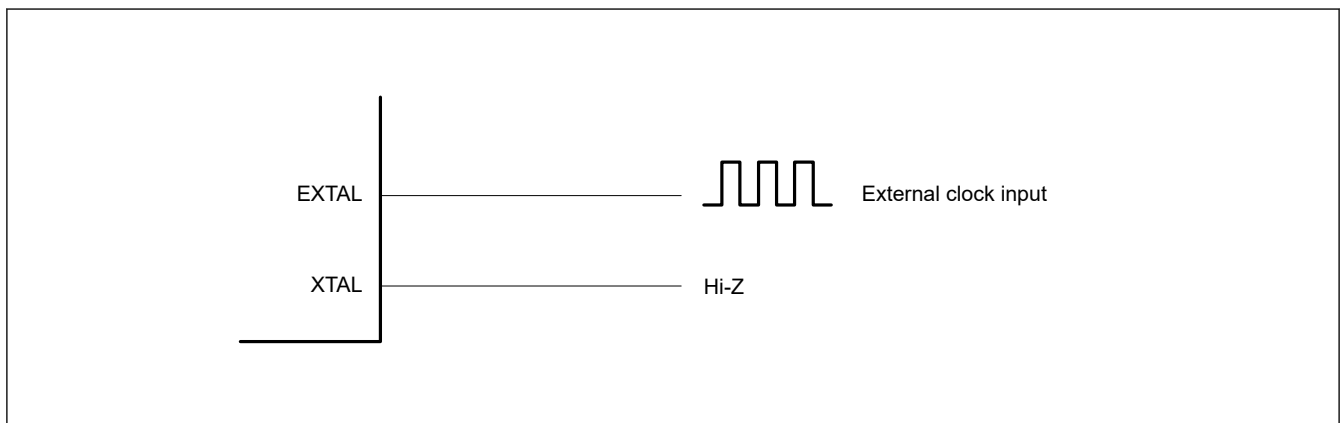


Figure 8.4 Equivalent circuit for external clock

### 8.3.3 Notes on External Clock Input

- The frequency of the external clock input can only be changed when the main clock oscillator is stopped. Do not change the frequency of the external clock input when the setting of the Main Clock Oscillator Stop bit (MOSCCR.MOSTP) is 0.
- To reduce current consumption during software standby mode, it is recommended to stop the external clock and keep it pull-up or pull-down.

However, if an external clock is used as the system clock source, it must be supplied while transitioning to and returning from software standby mode.

To do this safely, it is recommended to change the system clock source to another clock source before transitioning to software standby mode, or continue to use the external clock despite current consumption increase.

### 8.3.4 Note on Oscillation Keep in Software Standby mode

By using the oscillation keep function of the Main Clock Oscillator in Software Standby mode, the Main Clock Oscillator stabilization wait time after canceling Software standby mode can be shortened.

If MCU enters Software Standby mode under the MOSCSCR.MOSCSOKP bit is set to enabled (MOSCSCR.MOSCSOKP bit is 1) and under the Main Clock Oscillator is operating, the Main Clock Oscillator will keep oscillating in Software Standby mode. When canceling Software Standby mode in this state, the Main clock can be used without the oscillator stabilization wait time.

If using this function, confirm that the setting value is written to the register before entering the Software standby mode.

However, even if this function is enabled, the clock supply to the system and peripheral modules during Software Standby mode is stopped.

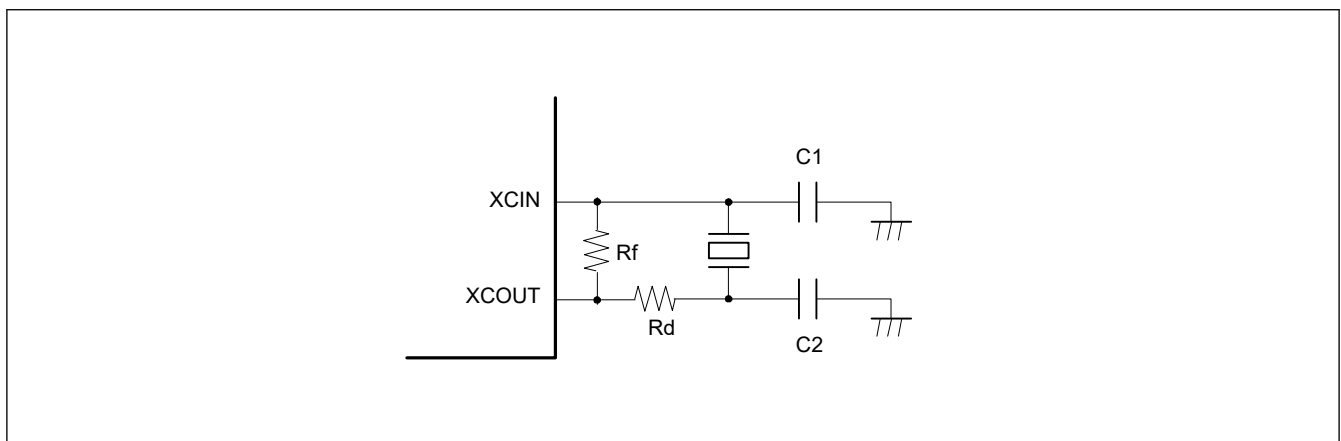
## 8.4 Sub-Clock Oscillator

To supply the clock signal to the sub-clock oscillator, use one of the following ways:

- Connect an oscillator
- Connect the input of an external clock signal.

### 8.4.1 Connecting a 32.768-kHz Crystal Resonator

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal resonator as shown in [Figure 8.5](#). A damping resistor ( $R_d$ ) can be added, if necessary. Because the resistor values vary according to the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If the resonator manufacturer recommends the use of an external feedback resistor ( $R_f$ ), insert an  $R_f$  between XCIN and XCOU by following the instructions. When connecting a resonator to supply the clock, the frequency of the resonator must be in the frequency range of the resonator for the sub-clock oscillator as described in [Table 8.1](#).



**Figure 8.5 Connection example of 32.768-kHz crystal resonator**

[Figure 8.6](#) shows an equivalent circuit for the 32.768-kHz crystal resonator.

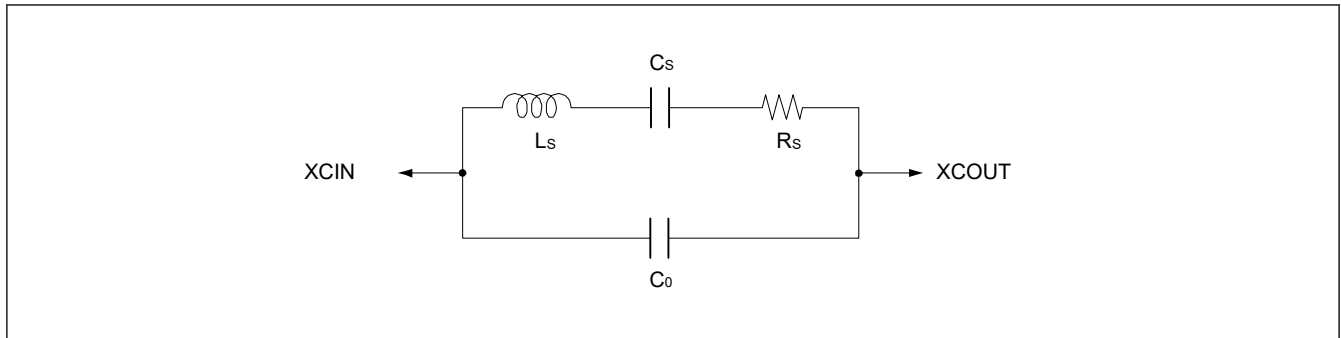


Figure 8.6 Equivalent circuit for the 32.768-kHz crystal resonator

### 8.4.2 Pin Handling When the Sub-Clock Oscillator Is Not Used

When the sub-clock oscillator is not in use, connect the XCIN pin to VSS through a resistor (to pull VSS down) and leave the XCOU pin open as shown in Figure 8.7. In addition, if an oscillator is not connected, set the Sub-Clock Oscillator Stop bit (SOSCCR.SOSTP) to 1 to stop the oscillator.

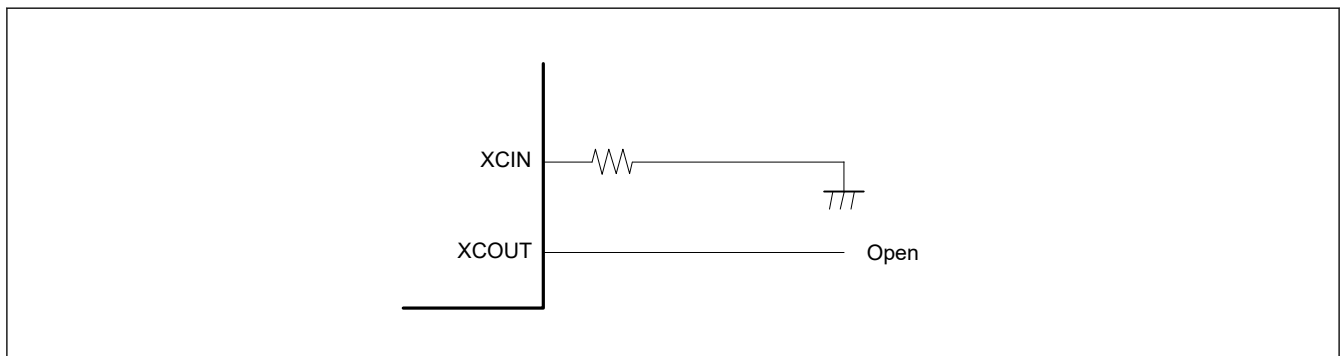


Figure 8.7 Pin handling when the sub-clock oscillator is not used

### 8.4.3 External Clock Input

In the external clock input mode, clock is input from EXCIN pin. In this case, set the XCIN/XCOU pin as shown in as Figure 8.7.

## 8.5 Low speed On Chip Oscillator (LOCO)

Low speed On Chip Oscillator (LOCO) is the internal oscillator that supplies a low-speed clock.

### 8.5.1 LOCO Operating conditions

There are restriction on LOCO operation setting. For detail, see [section 8.5.3. Restrictions on LOCO operation and stopping](#)

LOCO oscillates when any of the following conditions is met.\*1

[LOCO operating conditions]

- When LOCO is set to operate in the LOCOCR register (LOCOCR.LCSTP bit is 0).
- When IWDT is set to the auto start mode. (OFS0.IWDTSTRT bit is 0)\*2
- When IWDT is set to the register start mode (OFS0.IWDTSTRT bit is 1) and counting is started.\*3
- \When waiting for the oscillation stabilization of the Main Clock Oscillator (while MOCOCR.MCSTP bit is 0 and OSCSF.MOSCSF bit is 0).\*4

Note 1. LOCO stops at Deep Software Standby mode 2 or Deep Software Standby mode 3.

Note 2. When IWDT is set to the auto start mode, regardless of the setting of IWDTSTPCTL bit in OFS0 register ,LOCO continues to oscillate.

Note 3. When IWDT is set to the register start mode, regardless of the setting of SLCSTP bit in IWDTCSTPR register ,LOCO continues to oscillate once IWDT starts counting.

Note 4. Even if the Main Clock Oscillator is set to the external clock input mode, LOCO operates because the oscillation stabilization wait circuit operates when the Main Clock Oscillator starts operating.

### 8.5.2 LOCO Stop conditions

There are restriction on LOCO stop setting. For detail, see [section 8.5.3. Restrictions on LOCO operation and stopping](#).

There are two ways to top LOCO: stop by transition to Low Power mode or stop by LOCOCR register.

(A) LOCO stop conditions related to Low Power mode

LOCO stops when transitioning to the following Low Power mode

- Deep Software Standby mode 2
- Deep Software Standby mode 3

(B) LOCO stop conditions related to LOCOCR register

When LOCO is stopped using the LOCOCR register, even if the LCSTP bit set to stop (LOCOCR.LCSTP bit is 1), LOCO may not be stopped depending on the operating status of IWDT and oscillator.

LOCO can be stopped when all of the following conditions are met.

- LOCO is set to stop using the LOCOCR register (LOCOCR.LCSTP bit is 1)
- IWDT is not operating. (Include the case of stopping in CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode, and Deep Software Standby mode 1 by setting the OFS0.IWDTSTPCTL bit or the IWDTCSSTPR.SLCSTP bit.)
- Main Clock Oscillator is stopped or Main Clock oscillation is stable. (MOCOCR.MCSTP bit is 1 or OSCSF.MOSCSF bit is 1)

### 8.5.3 Restrictions on LOCO operation and stopping

After setting the LCSTP bit in LOCOCR to 0 to start the LOCO clock, only use the clock after the LOCO clock oscillation stabilization wait time (tLOCOWT) elapses. A fixed stabilization wait is required after setting the LOCO clock to start operation. A fixed wait for the oscillation to stop is also required.

The following restrictions apply when starting and stopping operation:

- After stopping the LOCO clock, read OSCMONR.LOCOMON, make sure LOCO is stopped, and then set the operation.
- Confirm that LOCO oscillation is stable before stopping the LOCO clock
- Confirm that LOCO oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or Deep Software Standby mode.
- When a transition to Software Standby mode or Deep Software Standby mode after setting LOCO to stop, read OSCMONR.LOCOMON, make sure LOCO is stopped, and executing the WFI instruction.

The LOCO clock is used to measure the oscillation stabilization wait time of the main clock oscillator. While the main clock oscillator is waiting for oscillation stabilization, the LOCO clock oscillates regardless of the LCSTP bit setting. At this time, it is necessary to prevent conflicts between the LOCO stop (or operation) by the LCSTP bit in LOCOCR and the LOCO clock oscillation for the Main clock Oscillator stabilization wait time (or LOCO clock stop after Main clock oscillation stabilization). Therefore, do not change the LOCOCR.LCSTP bit during the period from the 3 LOCO clock cycle before setting the start of main clock oscillator operation until the main clock oscillator stabilization can be confirmed with the oscillation stabilization flag in the OSCSF register.

When using IWDT in the register start mode (OFS0.IWDTSTRT bit is 1), set the LOCO to run (LOCOCR.LCSTP = 0) before IWDT count operation, and use IWDT after LOCO oscillation has stabilized. When IWDT is used in register start mode, do not change the LOCOCR.LCSTP bit during the period from the 3 LOCO clock cycles before IWDT count operation starts until the start of IWDT count operation can be confirmed.

When using the Digital filter or the Reset negate select for Voltage monitor m (m = 1 to 2) in Programmable Voltage Detection (PVDm), set the LOCO to run (LOCOCR.LCSTP = 0) before enabling PVDm and use PVDm after LOCO oscillation has stabilized.

## 8.6 Middle speed On Chip Oscillator (MOCO)

Middle speed On Chip Oscillator (MOCO) is the internal oscillator that supplies a middle-speed clock.

### 8.6.1 MOCO Operating conditions

There are restriction on MOCO operation setting. For detail, see [section 8.6.3. Restrictions on MOCO operation and stopping](#).

MOCO oscillates when any of the following conditions is met.

[MOCO operating conditions]

- When MOCO is set to operate in the LOCOCR register (MOCOCCR.MCSTP = 0)
- When waiting for the HOCO oscillation stabilization (while OFS1(\_SEC).HOCOEN bit is 0 or HOCOCCR.HCSTP bit is 0, and OSCSF.HOCOSF bit is 0).
- When waiting for the PLL1 oscillation stabilization (while PLLCR.PLLSTP bit is 0, and OSCSF.PLLSF bit is 0).
- When waiting for the PLL2 oscillation stabilization (while PLL2CR.PLL2STP bit is 0, and OSCSF.PLL2SF bit is 0).

Note: MOCO stops at Software Standby mode or Deep Software Standby mode.

### 8.6.2 MOCO Stop conditions

There are restriction on MOCO stop setting. For detail, see [section 8.6.3. Restrictions on MOCO operation and stopping](#).

There are two ways to top MOCO: stop by transition to Low Power mode or stop by MOCOCCR register.

(A) MOCO stop conditions related to Low Power mode

MOCO stops when transitioning to the following Low Power mode.

- Software Standby mode
- Deep Software Standby mode

Note: When the On-Chip Debugger function is enabled, MOCO does not stop when transitioning to each standby mode.

(B) MOCO stop conditions related to MOCOCCR register

When MOCO is stopped using the MOCOCCR register, even if the MCSTP bit set to stop (MOCOCCR.MCSTP bit is 1), MOCO may not be stopped depending on the operating status of the system.

MOCO can be stopped when all of the following conditions are met.

[MOCO stop conditions]

- MOCO is set to stop using the MOCOCCR register (MOCOCCR.MCSTP bit is 1)
- When HOCO is stopped or HOCO oscillation is stable. (OFS1(\_SEC).HOCOEN bit is 1 and HOCOCCR.HCSTP bit is 1, or OSCSF.HOCOSF bit is 1)
- PLL1 circuit is stopped or PLL1 oscillation is stable. (PLLCR.PLSTP bit is 1 or OSCSF.PLLSF bit is 1)
- PLL2 circuit is stopped or PLL2 oscillation is stable. (PLL2CR.PL2STP bit is 1 or OSCSF.PLL2SF bit is 1)

### 8.6.3 Restrictions on MOCO operation and stopping

After setting the MCSTP bit to 0 to start the MOCO clock, only use the clock after the MOCO clock oscillation stabilization time (tMOCOWT) elapses. A fixed stabilization wait is required after setting the MOCO clock to start operation. A fixed wait for oscillation to stop is also required.

The following restrictions apply when starting and stopping operation.

- After stopping the MOCO clock, read OSCMONR.MOCOMON, make sure MOCO is stopped, and then set the operation.
- Confirm that MOCO oscillation is stable before stopping the MOCO clock.
- Regardless of whether the MOCO clock is selected as the system clock source, confirm that MOCO oscillation is stable before executing a WFI instruction to place the MCU in Software Standby or CPU Deep Software Standby mode. If MOCO is stopped, operate the MOCO.

- Regardless of whether the MOCO clock is selected as the system clock source, confirm that MOCO oscillation is stable before executing the wait instruction to place the CPU in the CPU Deep Sleep mode. If MOCO is stopped, operate the MOCO.
- Confirm that MOCO oscillation is stable before programming or erasing Flash Memory. If MOCO is stopped, operate the MOCO.

Writing 1 to MCSTP is prohibited under the following condition:

- SCKSCR.CKSEL[2:0] = 001b (system clock source = MOCO).
- Oscillation stop detection is enabled (OSTDCR.OSTDE bit is 1).
- While waiting for the HOCO oscillation stabilization (OFS1(\_SEC).HOCOEN bit is 0 or HOCOEN.HCSTP bit is 0, and OSCSF.HOCOSF bit is 0)
- While waiting for the PLL1 oscillation stabilization (PLLCR.PLLSTP bit is 0, and OSCSF.PLLSF bit is 0)
- While waiting for the PLL2 oscillation stabilization (PLL2CR.PLL2STP bit is 0, and OSCSF.PLL2SF bit is 0)
- While measuring the Operation Stability wait time for Flash Memory
- While Flash Memory programming or erasure

The MOCO clock is used to measure the oscillation stabilization wait time of the HOCO/PLL1/PLL2 clock. While waiting for the HOCO/PLL1/PLL2 clock to stabilize, the MOCO clock oscillates regardless of the MOCOEN.MCSTP bit setting. At this time, it is necessary to prevent conflicts between the MOCO stop (or operation) by the MCSTP bit in MOCOEN and the MOCO clock oscillation for HOCO/PLL1/PLL2 clock oscillation stabilization wait time (or MOCO clock stop after HOCO/PLL1/PLL2 clock oscillation stabilization). Therefore, do not change the MOCOEN.MCSTP bit during the period from the 3 MOCO clock cycles before the start of HOCO/PLL1/PLL2 clock operation setting until the HOCO/PLL1/PLL2 clock oscillation stability can be confirmed by the HOCOSF/PLLSF/PLL2SF flag in the OSCSF register.

## 8.7 High speed On Chip Oscillator (HOCO)

High-speed On Chip Oscillator (HOCO) is the internal oscillator that supplies a high-speed clock.

### 8.7.1 Frequency Locked Loop Function

HOCO supports the Frequency Locked Loop (FLL) function that corrects the frequency accuracy of the HOCO clock using the Sub-Clock Oscillator. When the FLL function is enabled, the frequency accuracy of the HOCO clock can be improved compared to when the FLL function is disabled.

When using the FLL function, the oscillation of the Sub-Clock Oscillator must be stable. In addition, the sub-clock oscillator must meet the frequency accuracy specified in the electrical characteristics. (For details, [section 52, Electrical Characteristics](#) see section Electrical Characteristics)

When enabling the FLL function and using HOCO, enable the FLL function in advance and enable HOCO (HOCOEN.HCTSP=0) after SOSC has stabilized.

Set FLLCR2.FLLCNTL[10:0] as follows for each HOCO frequency setting. And the HOCO clock reaches the desired frequency after tFLLWT after enabling HOCO.

**Table 8.5 FLLCNTL setting for each HOCO frequency setting**

FLLCR2.FLLCNTL[10:0]	OFS1(_SEC).HOCOFREQ[2:0]	Target Frequency (FHOCO)
0x1E9	000b	16 MHz
0x226	001b	18 MHz
0x263	010b	20 MHz
0x1E9	100b	32 MHz
0x1E9	111b	48 MHz

### 8.7.2 Note on Oscillation Keep in Software Standby mode

By using the oscillation keep function of the HOCO in Software Standby mode, the HOCO oscillation stabilization wait time after canceling Software standby mode can be shortened.

To keep the HOCO oscillation in Software Standby mode, make the following settings:

- Enable the HOCO Standby Oscillation Keep function in the HOCOSCR register. (HOCOSCR.HOCOSOKP bit is set to 1)
- The setting of power control for HOCO in HOCOLDOCR.

If MCU enters Software Standby mode under the above settings and under the HOCO is operating, the HOCO will keep oscillating in Software Standby mode. When canceling Software Standby mode in this state, the HOCO clock can be used without the oscillator stabilization wait time.

If using this function, confirm that the setting value is written to the register before entering the Software standby mode.

However, even if this function is enabled, the clock supply to the system and peripheral modules during Software Standby mode is stopped.

### 8.7.3 Note on using as TRACE clock source

If HOCO is selected as Trace clock source with TRCKCRSEL bit on TRCKCR register, and then Trace clock function is enable (TRCKCR.TRCKEN=1), HOCO oscillates regardless of the value of HCSTP bit on HOCOCR register. In this situation, HOCO will continue to oscillate even if MCU is in Software Standby mode, Deep Software Standby mode, or Reset (except POR and PVD0) to keep supplying the clock to the TRACE module.

If HOCO is used as Trace clock source, Trace clock should be enabled with TRCKEN bit on TRCKCR register while HOCO is stopped (OFS1(\_SEC).HOCOEN=1 and HOCOCR.HCSTP=1) or HOCO is oscillating and stable (OSCSF.HOCOSF=1).

## 8.8 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

[Figure 8.8](#) shows the block diagram of PLL1 circuit, and [Figure 8.9](#) shows the PLL2 circuit. PLL1 and PLL2 have the same circuit configuration. The PLL1 circuit is explained the following as an example.

The input clock source selection circuit selects the clock source input to the PLL.

The input clock divider circuit divides the input clock and generates the PLL reference clock.

The frequency control circuit controls the VCO clock to be generated from the PLL reference clock according to the frequency multiplier setting. The frequency multiplication ratio is determined by the combination of the PLLMUL bits and PLLMULNF bits. The PLLMUL bit determines the integer part of the multiplication ratio, and the PLLMULNF bit determines the fractional part of the multiplication ratio. For example, if the PLLMUL bit setting is 26 and the PLLMULNF bit setting is 0.33, the multiplication ratio is 26.33.

The PLL circuit has the 3 output clocks. The output clock divider circuit divides the VCO clock to generate each PLL clock output. The division ratio for generating each output clock can be set independently with the PLODIVP, PLODIVQ, and PLODIVR bits.

The PLL Input clock frequency, the PLL Reference clock frequency, the VCO clock frequency, and each Output clock frequency must be within the specified in [section 52, Electrical Characteristics](#).

The PLL frequency setting can only be changed the PLL circuit is stopped. Do not change the PLL frequency setting when PLL stop control bit (PLLSTP) is 0.



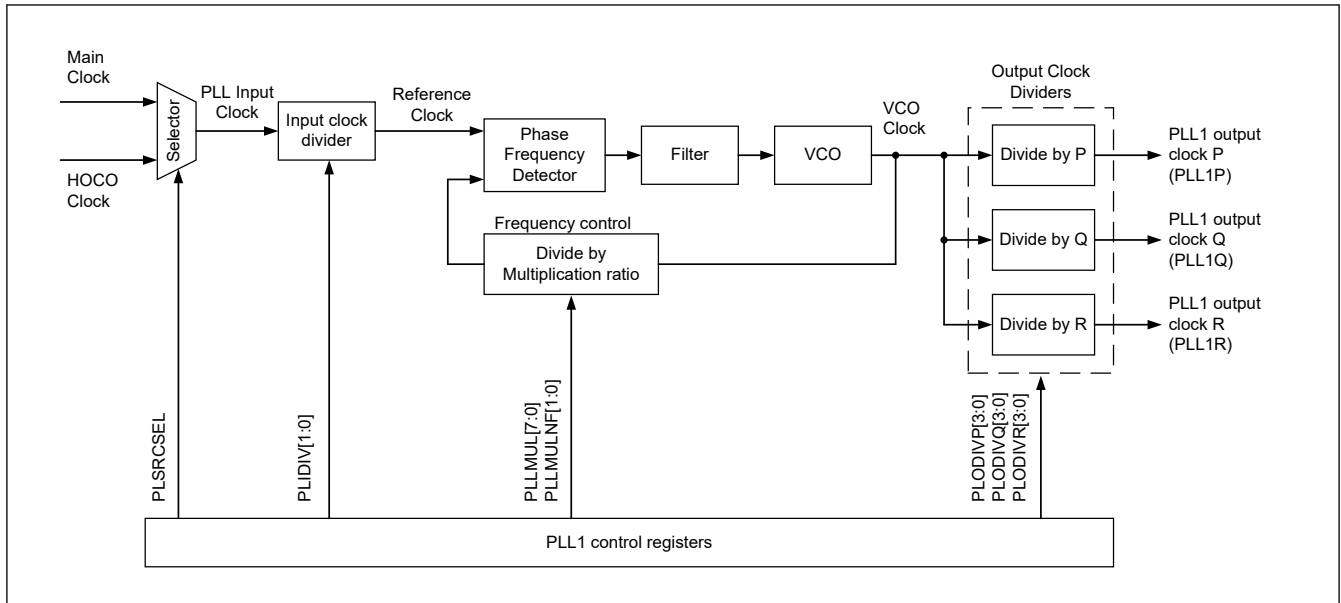


Figure 8.8 PLL1 block diagram

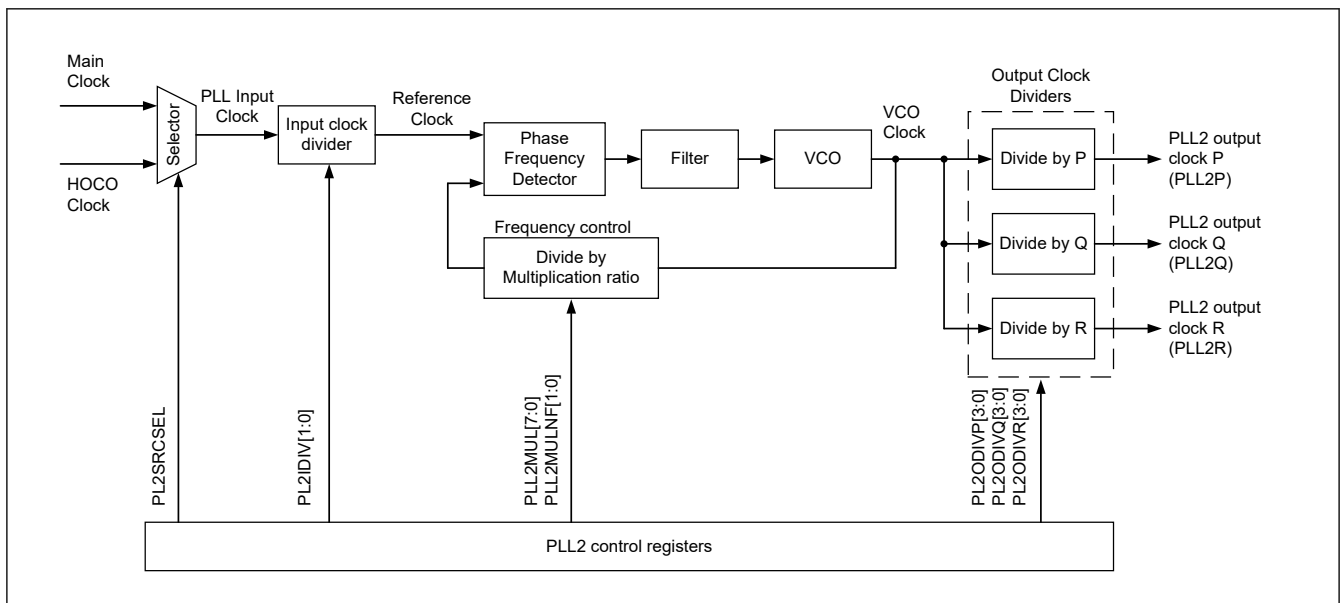


Figure 8.9 PLL2 block diagram

## 8.9 Oscillation Stop Detection Function

### 8.9.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function detects the main clock oscillator stop. When oscillation stop is detected, the system clock source switches as follows:

- If an oscillation stop is detected with  $SCKSCR.CKSEL[2:0] = 011b$  (system clock source = MOSC), the system clock source switches to the MOCO clock.
- If an oscillation stop is detected with  $PLLCCR.PLSRCSEL = 0$  (PLL1 source clock = MOSC) and  $SCKSCR.CKSEL[2:0] = 101b$  (system clock source = PLL1P), PLL1 clock remains the system clock source. The frequency becomes free-running, and the setting in the  $SCKSCR.CKSEL[2:0]$  bits does not change.

An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the General PWM Timer (GPT) output can be forced to stop state on detection.

The main clock oscillation stop is detected when the input clock remains at 0 or 1 for a certain period, for example, when a malfunction occurs in the main clock oscillator. See [section 52, Electrical Characteristics](#).

Switching between the main clock oscillator and the MOCO clock or between the PLL1 clock and PLL1 free-running clock is controlled by the Oscillation Stop Detection Flag (OSTDSR.OSTDF).

OSTDF controls the switched clock as follows:

- When SCKSCR.CKSEL[2:0] = 011b (system clock source = MOSC):
  - When OSTDF changes from 0 to 1, the clock source switches to the MOCO clock.
  - When OSTDF changes from 1 to 0, the clock source switches back to MOSC.
- When PLLCCR.PLSRCSEL = 0 (PLL1 source clock = MOSC) and SCKSCR.CKSEL[2:0] = 101b (System clock source = PLL1P):
  - When OSTDF changes 0 to 1, the clock source switches to the PLL1 free-running oscillation clock.
  - When OSTDF changes 1 to 0, the clock source switches back to PLL1.

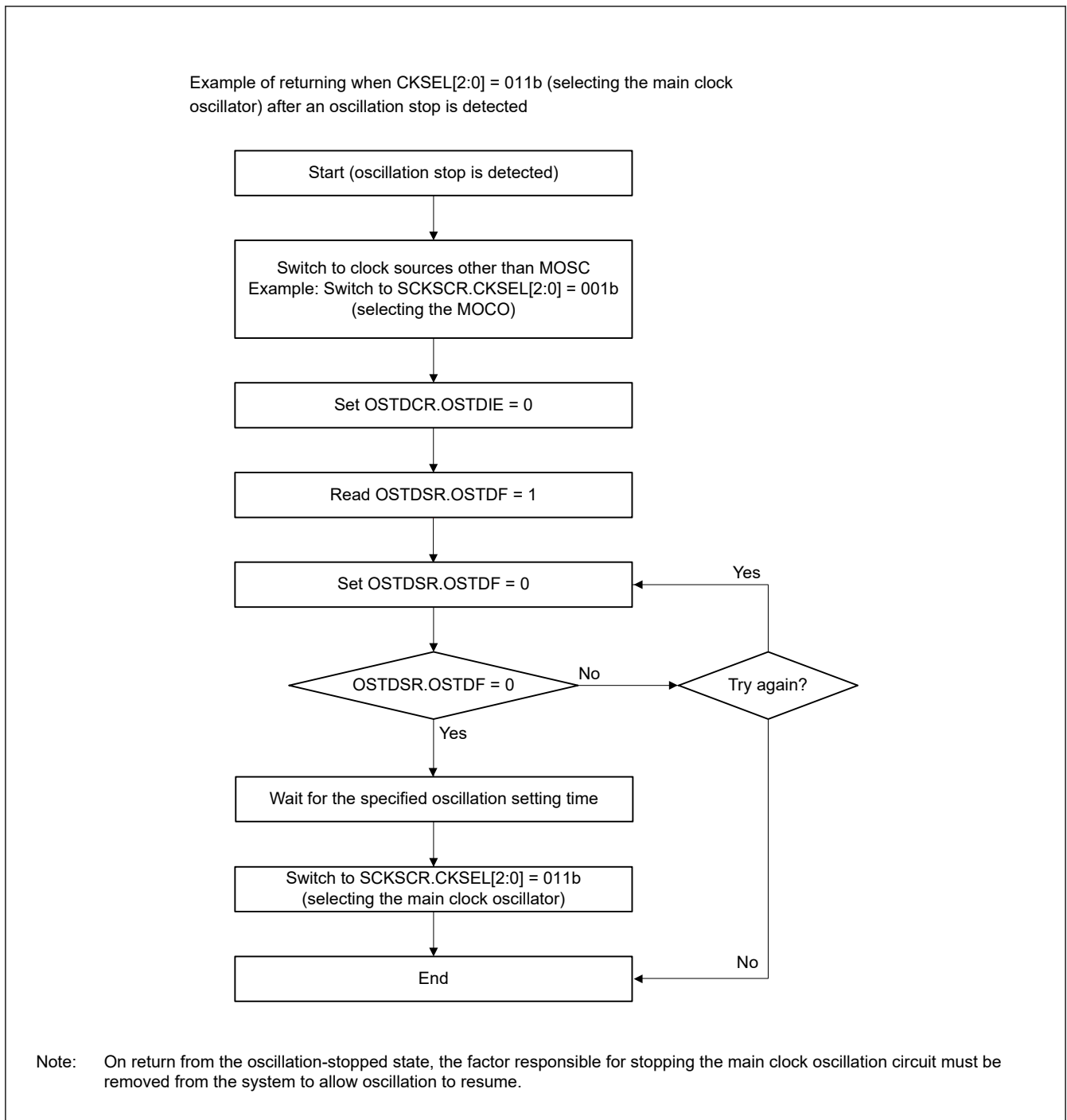
To switch the clock source to the main clock or PLL1 clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL1 clock and clear the OSTDF flag to 0. Also, check that the OSTDF flag is not 1, then set the CKSEL[2:0] bits to the main clock or PLL1 clock after the specified oscillation stabilization time elapses.

After a reset release, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after a specified oscillation stabilization time elapses.

The oscillation stop detection function detects when the main clock is stopped by an external cause.

The oscillation stop detection function switches the following clocks to the MOCO clock (when the system clock source is MOSC) or the PLL1 free-running clock (when the system clock source is PLL1P) :

- All clocks that can be selected as the MOSC clock or PLL1 except CLKOUT and CACMCLK.
- The system clock (ICLK) frequency during the MOCO (when the system clock source is MOSC) or PLL1 free-running (when the system clock source is PLL1P) operation is specified by the MOCO oscillation frequency and the division ratio set by the system clock select bits (SCKDIVCR.ICK[3:0])



**Figure 8.10** Flow of recovery on detection of oscillator stop

### 8.9.2 Oscillation Stop Detection Interrupts

An oscillation stop detection interrupt (MOSC\_STOP) is generated when the Oscillation Stop Detection Flag (OSTDSR.OSTDF) is 1 and the Oscillation Stop Detection Interrupt Enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDIE) is 1 (enabled). The Port Output Enable for GPT (POEG) is notified of the main clock oscillator stop. On receiving the notification, the POEG sets the Oscillation Stop Detection Flag in the POEG Group n Setting Register (POEGGn.OSTPF) to 1 (n = A, B, C, D).

After the oscillation stop is detected, wait at least 10 PCLKB clock cycles before writing to the POEGGn.OSTPF flag. When the OSTDSR.OSTDF flag requires clearing, do so after clearing the Oscillation Stop Detection Interrupt Enable bit in the Oscillation Stop Detection Control Register (OSTDCR.OSTDIE). Wait at least 2 PCLKB clock cycles before setting the OSTDCR.OSTDIE bit to 1 again. A longer PCLKB wait time might be required, depending on the number of cycles required to read a given I/O register.

The oscillation stop detection interrupt is a non-maskable interrupt. Because non-maskable interrupts are disabled in the initial state after a reset release, enable non-maskable interrupts through software before using oscillation stop detection interrupts. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

## 8.10 Internal Clock

Clock sources for the internal clock signals include:

- Main clock oscillator
- Sub-clock oscillator
- HOCO clock
- MOCO clock
- LOCO clock
- PLL1 clock (PLL1P, PLL1Q and PLL1R)
- PLL2 clock (PLL2P, PLL2Q and PLL2R)
- External clock input for JTAG
- External clock input for SWD

The following internal clocks are produced from these sources.

- Operating clock of the CPU : CPU clock (CPUCLK)
- Operating clock of the DMAC, DTC, Flash, SRAM, System Bus , I/O Port and ICU: System clock (ICLK)
- Operating clock of the Debug Subsystem : Debug clock (DCLK)
- Operating clocks of peripheral modules: Peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD, and PCLKE)
- Operating clock of the FlashIF: FlashIF clock (FCLK)
- Clock for the trace function and external pin output : Trace clock (TRCLK)
- Operating clock for the SCI : SCI clock (SCICLK)
- Operating clock for the SPI : SPI clock (SPICLK)
- Operating clock for the Octal SPI : Octal-SPI clock (OCTACLK, OCTADIVCLK)
- Operating clock for the CANFD Core : CANFD Core clock (CANFDCLK)
- Operating clock for the USBFS : USB clock (USBCLK)
- Clock for external pin output : Clock/Buzzer output (CLKOUT)
- Operating clock for the CAN : CAN clock (CANMCLK)
- Operating clocks for the ULPT : ULPT LOCO clock (ULPTLCLK) and ULPT sub-clock (ULPTSCLK)
- Operating clock for the AGT : AGT LOCO clock (AGTLCLK) and AGT sub-clock (AGTSCLK)
- Operating clocks for the CAC : CAC clock (CACCLK)
- Operating clocks for the RTC : RTC LOCO clock (RTCLCLK) and RTC Sub-clock (RTCSCLK)
- Operating clock for the IWDT : IWDT clock (IWDTCLK)
- Operating clock for the SysTick timer : SysTick timer clock (SYSTICKCLK)
- Operating clock for the JTAG : JTAG clock (JTAGTCK)

For details on the registers used to set the frequencies of the internal clocks, see [section 8.10.2. System Clock \(ICLK\)](#) to [section 8.10.20. JTAG Clock \(JTAGTCK\)](#)

If the value of any of these bits is changed, subsequent operation is at the frequency determined by the new value.

### 8.10.1 CPU Clock (CPUCLK)

The CPU clock (CPUCLK) is the operating clock for the CPU. Specify the frequency in the following bits:

- CPUCK[3:0] bits in SCKDIVCR2

- CKSEL[2:0] bits in SCKSCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0], and PLODIVP[3:0] bits in PLLCCR and PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(\_SEC)<sup>\*1</sup>

Note 1. OFS1 is for non-secure developers and OFS1\_SEC is for secure developers. The applied setting value is determined by OFS1\_SEL. The value of OFS1(\_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

### 8.10.2 System Clock (ICLK)

The system clock (ICLK) is the operating clock of the DMAC, DTC, Flash, SRAM, System Bus, I/O Port, and ICU. Specify the frequency in the following bits:

- ICK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(\_SEC)<sup>\*1</sup>

Note 1. OFS1 is for non-secure developers and OFS1\_SEC is for secure developers. The applied setting value is determined by OFS1\_SEL. The value of OFS1(\_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

When the ICLK clock source is switched, the duration of the ICLK clock cycle becomes longer during the clock source transition period. See [Figure 8.11](#) and [Figure 8.12](#).

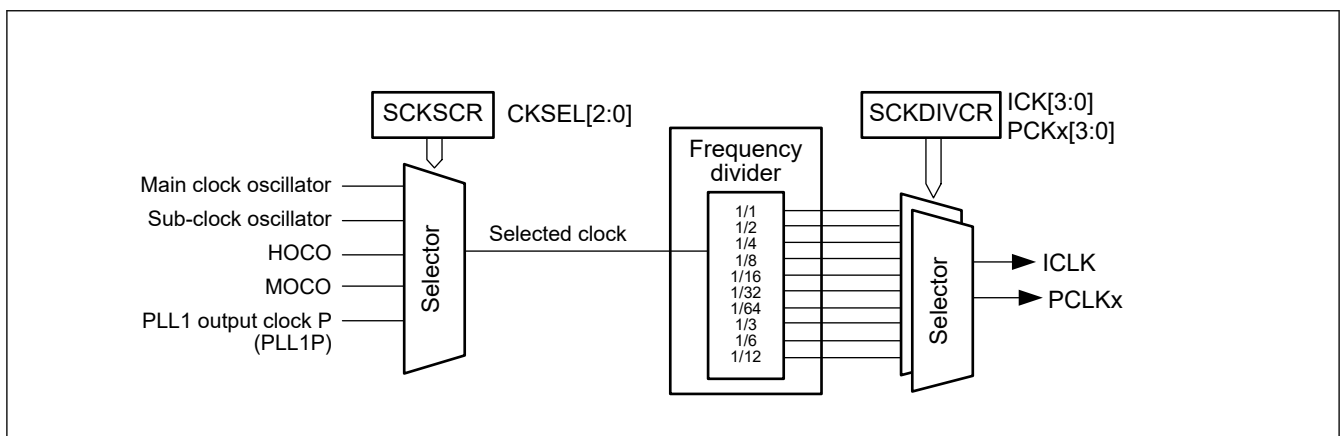
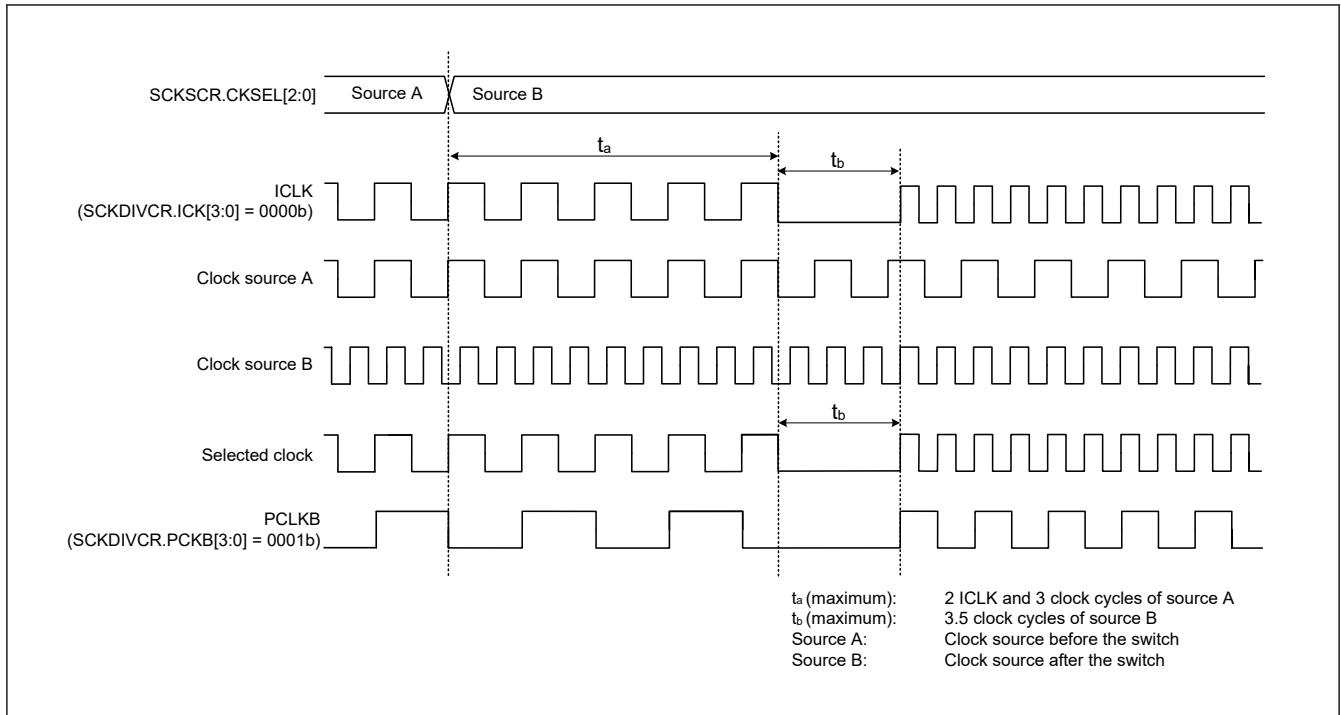


Figure 8.11 Block diagram of clock source selector



**Figure 8.12** Timing of clock source switching

### 8.10.3 Debugger Clock (DCLK)

The Debugger clock (DCLK) is the operating clock for the Debug Subsystem. The frequency is ICLK divided by 2.

### 8.10.4 Peripheral Module Clock (PCLKA, PCLKB, PCLKC, PCLKD, PCLK E)

The peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD and PCLK E) are the operating clocks for the peripheral modules.

The frequency of the given clock is specified in the following bits:

- PCKA[3:0], PCKB[3:0], PCKC[3:0], PCKD[3:0] and PCKE[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- HOCOFRQ0[2:0] bits in OFS1(\_SEC)\*1.

Note 1. OFS1 is for non-secure developers and OFS1\_SEC is for secure developers. The applied setting value is determined by OFS1\_SEL. The value of OFS1(\_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

When the clock source of the peripheral module clock is switched, the duration of the peripheral module clock cycle becomes longer during the clock source transition period. See [Figure 8.11](#) and [Figure 8.12](#).

### 8.10.5 FlashIF Clock (FCLK)

The flash interface clock (FCLK) is the operating clock for the flash memory interface. In addition to reading from the data flash, FCLK is used for the programming and erasure of the code flash and data flash.

The FCLK frequency is specified in the following bits:

- FCK[3:0] bits in SCKDIVCR
- CKSEL[2:0] bits in SCKSCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2

- HOCOFRQ0[2:0] bits in OFS1(\_SEC)\*1.

Note 1. OFS1 is for non-secure developers and OFS1\_SEC is for secure developers. The applied setting value is determined by OFS1\_SEL. The value of OFS1(\_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

### 8.10.6 Trace Clock (TRCLK)

The trace clock (TRCLK) is the CPU tracing clock for the On-Chip Debugger function.

The TRCLK frequency is specified by the TRCK[3:0] bits and TRCKSEL bit in TRCKCR (The Trace Clock frequency division ratio and the Trace Clock Source select) . When changing the frequency of the Trace Clock, the Trace Clock must be stopped (TRCKCR.TRCKEN = 0).

### 8.10.7 SCI Clock (SCICLK)

The SCI clock (SCICLK) is the operating clock for the SCI module.

Specify the frequency in the following bits:

- SCICKDIV[2:0] bits in SCICKDIVCR
- SCICKSEL[3:0] bits in SCICKCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(\_SEC)\*1.

Note 1. OFS1 is for non-secure developers and OFS1\_SEC is for secure developers. The applied setting value is determined by OFS1\_SEL. The value of OFS1(\_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

### 8.10.8 SPI Clock (SPICLK)

The SPI clock (SPICLK) is the operating clock for the SPI module.

Specify the frequency in the following bits:

- SPICKDIV[2:0] bits in SPICKDIVCR
- SPICKSEL[3:0] bits in SPICKCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(\_SEC)\*1.

Note 1. OFS1 is for non-secure developers and OFS1\_SEC is for secure developers. The applied setting value is determined by OFS1\_SEL. The value of OFS1(\_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

### 8.10.9 Octal-SPI clock (OCTACLK, OCTADIVCLK)

The Octal-SPI clock (OCTACLK) is the operating clock for the Octal-SPI module.

Specify the frequency in the following bits:

- OCTACKDIV[2:0] bits in OCTACKDIVCR
- OCTACKSEL[3:0] bits in OCTACKCR

- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(\_SEC)\*1.

Note 1. OFS1 is for non-secure developers and OFS1\_SEC is for secure developers. The applied setting value is determined by OFS1\_SEL. The value of OFS1(\_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

OCTADIVCLK always outputs 2-division of OCTACLK.

### 8.10.10 CANFD Core clock (CANFDCLK)

The CANFD Core clock (CANFDCLK) is the operating clock for the CANFD module.

Specify the frequency in the following bits:

- CANFDCKDIV[2:0] bits in CANFDCKDIVCR
- CANFDCKSEL[3:0] bits in CANFDCKCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(\_SEC)\*1.

Note 1. OFS1 is for non-secure developers and OFS1\_SEC is for secure developers. The applied setting value is determined by OFS1\_SEL. The value of OFS1(\_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

### 8.10.11 USB Clock (USBCLK)

The USB clock (USBCLK) is the operating clock for the USBFS module.

A 48-MHz clock must be supplied when using the USBFS module.

The USBCLK frequency is specified in the following bits:

- USBCKDIV[2:0] bits in USBCKDIVCR
- USBCKSEL[3:0] bits in USBCKCR
- PLLMUL[7:0], PLLMULNF[1:0], PLIDIV[1:0], PLODIVP[3:0], PLODIVQ[3:0] and PLODIVR[3:0] bits in PLLCCR and PLLCCR2
- PLL2MUL[7:0], PLL2MULNF[1:0], PL2IDIV[1:0], PL2ODIVP[3:0], PL2ODIVQ[3:0] and PL2ODIVR[3:0] bits in PLL2CCR and PLL2CCR2
- HOCOFRQ0[2:0] bits in OFS1(\_SEC)\*1.

Note 1. OFS1 is for non-secure developers and OFS1\_SEC is for secure developers. The applied setting value is determined by OFS1\_SEL. The value of OFS1(\_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

### 8.10.12 CAN Clock (CANMCLK)

The CAN clock (CANMCLK) is the operating clock for the CAN module. CANMCLK is generated by the main clock oscillator.

### 8.10.13 ULPT Clock (ULPTLCLK, ULPTSCLK)



The ULPT clocks (ULPTLCLK and ULPTSCLK) are the operating clock for the ULPT module. ULPTLCLK is generated by the LOCO clock and ULPTSCLK is generated by the Sub-clock oscillator.

#### 8.10.14 CAC Clock (CACCLK)

The CAC clock (CACCLK) is the operating clock for the CAC. CACCLK includes the following clocks:

- CAC Main clock (CACMCLK) is generated by the Main clock oscillator
- CAC Sub oscillator clock (CACSCLK) is generated by the Sub-clock oscillator
- CAC HOCO clock (CACHCLK) is generated by the HOCO clock
- CAC MOCO clock (CACMOCLK) is generated by the MOCO clock
- CAC LOCO clock (CACLCLK) is generated by the LOCO clock

#### 8.10.15 RTC Clock (RTCSCLK, RTCLCLK)

The RTC clock (RTCSCLK, RTCLCLK) is the operating clock for the RTC.

RTCSCLK is generated by the sub-clock oscillator, and RTCLCLK is generated by the LOCO clock.

#### 8.10.16 IWDTClock (IWDTCLK)

The IWDTClock (IWDTCLK) is the operating clock for the IWDTClock.

IWDTCLK is internally generated by the LOCO clock which is always divided by 2.

#### 8.10.17 AGT Clock (AGTSCLK, AGTLCLK)

The AGT clocks (AGTSCLK and AGTLCLK) are the operating clocks for the AGT. AGTSCLK is generated by the sub-clock oscillator, and AGTLCLK is generated by the LOCO clock.

#### 8.10.18 SysTick Timer Clock (SYSTICKCLK)

The SysTick timer clock (SYSTICKCLK) is the operating clock for the SysTick timer. SYSTICKCLK is generated by the MOCO clock which is always divided by 8.

#### 8.10.19 External Pin Output Clock (CLKOUT)

The CLKOUT is output externally from the CLKOUT pin for the clock or buzzer output. The CLKOUT is output to the CLKOUT pin when the CKOCR.CKOEN bit is set to 1. Only change the value in the CKODIV[2:0] bits or CKOSEL[2:0] bits in CKOCR when the CKOCR.CKOEN bit is 0.

The CLKOUT clock frequency is specified in the following bits:

- CKODIV[2:0] bits or CKOSEL[2:0] bits in CKOCR
- HOCOFRQ0[2:0] bits in OFS1(\_SEC)\*<sup>1</sup>

Note 1. OFS1 is for non-secure developers and OFS1\_SEC is for secure developers. The applied setting value is determined by OFS1\_SEL. The value of OFS1(\_SEC).HOCOFRQ0[2:0] bits is automatically transferred to HOCOCR2.HCFRQ0[2:0] bits after reset, therefore HOCO frequency can also be specified by HOCOCR2.HCFRQ0[2:0] bits.

#### 8.10.20 JTAG Clock (JTAGTCK)

The JTAG clock (JTAGTCK) is the clock for the JTAG.

JTAGTCK is generated by the JTAG external clock (TCK).

### 8.11 Clock Setting

#### 8.11.1 System Clock Setting

### 8.11.1.1 Initial System Clock Setting

An example of the initial system clock setting procedure is shown in [Table 8.6](#).

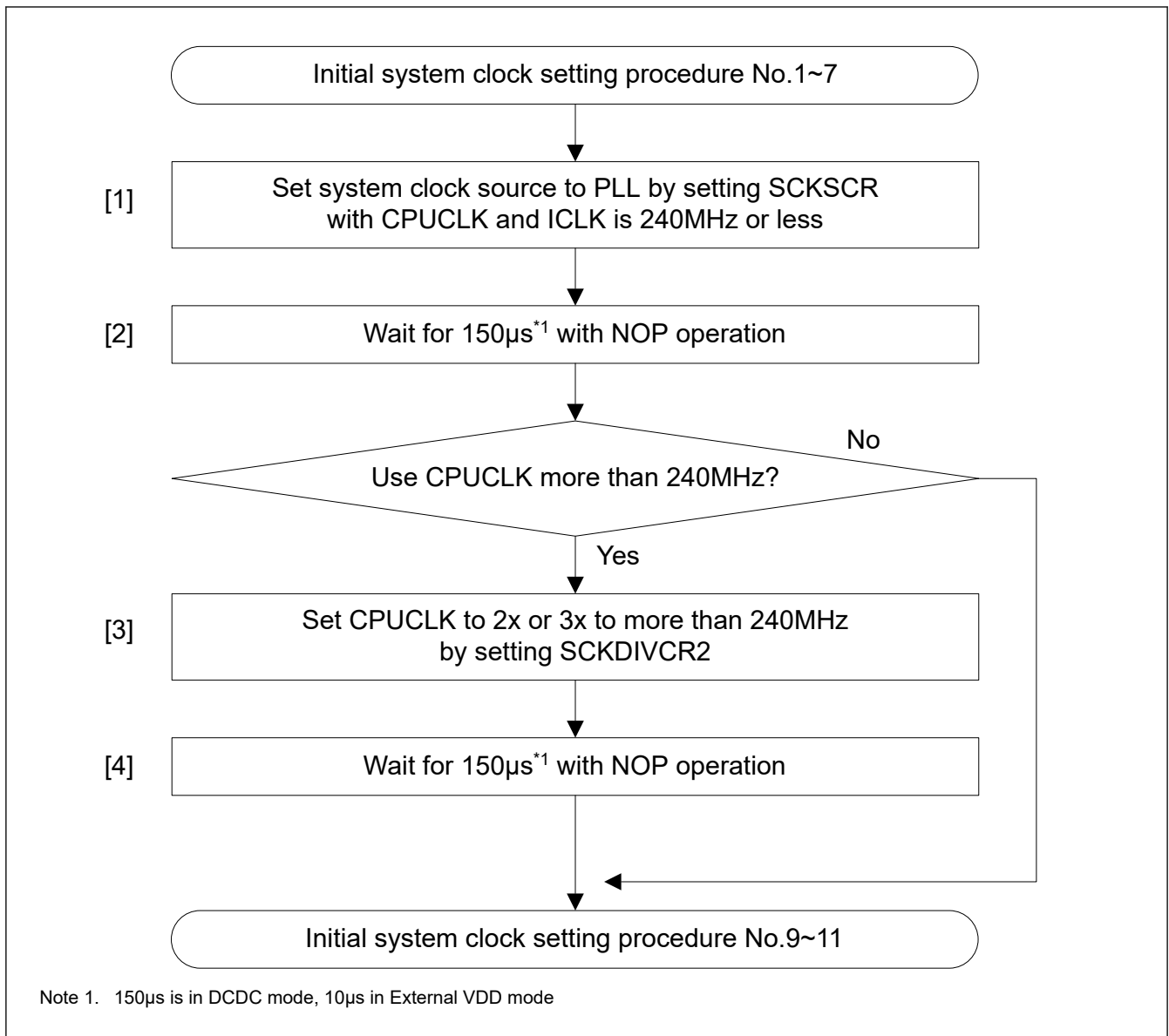
**Table 8.6 Example of the initial system clock setting procedure after reset release / after Deep Software Standby cancellation**

No.	Step	Description
1	Start	After reset release / Deep software standby cancellation (MOCO is selected as system clock source)
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 <sup>*1</sup> bit in PRCR register
3	Change Operating Power Control Mode to High-Speed mode <sup>*1</sup>	Change to High-Speed mode with OPCCR register. <sup>*1</sup> For details, see <a href="#">section 10, Low Power Modes</a> .
4	Set a clock source to oscillate	Set a clock source that switches to the system clock source to oscillate.
5	Set Flash Wait Cycle	Set an appropriate access wait count for the flash memory according to the frequency at which it operates after the system clock (ICLK) is set. For details, see <a href="#">section 49, Flash Memory</a> .
6	Set internal clock division ratio	Set the internal clock division ratio with SCKDIVCR register.
7	Switch the System Clock source	Switch the system clock source with SCKSCR register. Before switching the system clock source, confirm that the clock source oscillation is stable.
8	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. <sup>*2</sup>
9	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 <sup>*1</sup> bit in PRCR register
10	End	System clock setting is completed.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.

Note 2. Setting is not necessary if the operating power control mode is not changed.

When the PLL1 is selected as the system clock source, additional procedure is required for procedure No.8. This additional procedure is shown in [Figure 8.13](#). It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt.



**Figure 8.13** Additional flow of initial system clock setting when PLL is used for the system clock source. (assuming all module stop bis are initial value)

### 8.11.1.2 System Clock Setting for Faster Frequency

An example of setting procedure when changing the current system clock frequency to a faster frequency is shown in [Table 8.7](#)

**Table 8.7** Example of Setting Procedure When changing the current system clock frequency to a Faster frequency (1 of 2)

No.	Step	Description
1	Start	
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 <sup>1</sup> 2 bit in PRCR register
3	Change Operating Power Control Mode to High-Speed mode <sup>*1</sup>	Change to High-Speed mode with OPCCR register. <sup>*1</sup> For details, see <a href="#">section 10, Low Power Modes</a> .
4	Set a clock source to oscillate <sup>*3</sup>	Set a clock source that switches to the system clock source to oscillate. <sup>*4</sup>
5	Set Flash Wait Cycle	Set an appropriate access wait count for the flash memory according to the frequency at which it operates after the system clock (ICLK) is set. For details, see <a href="#">section 49, Flash Memory</a> .

**Table 8.7 Example of Setting Procedure When changing the current system clock frequency to a Faster frequency (2 of 2)**

No.	Step	Description
6	Set internal clock division ratio	Set the internal clock division ratio with SCKDIVCR register.
7	Switch the System Clock source <sup>*3</sup>	Switch the system clock source with SCKSCR register. Before switching the system clock source, confirm that the clock source oscillation is stable.
8	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. <sup>*2</sup>
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 <sup>*1*2</sup> bit in PRCR register
11	End	System clock is changed to a Faster frequency.

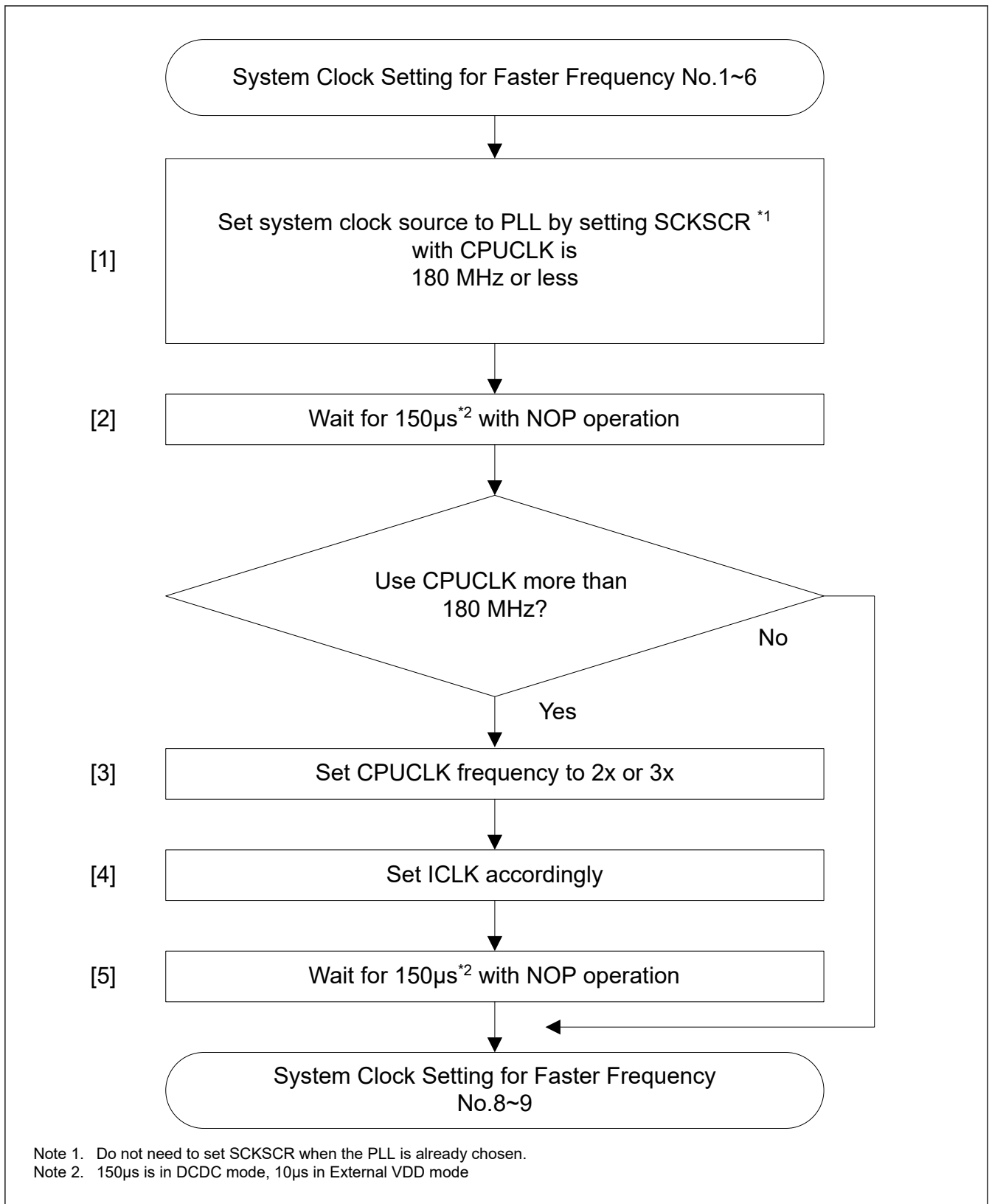
Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.

Note 2. Setting is not necessary if the operating power control mode is not changed.

Note 3. Setting is not necessary if the system clock source is not changed.

Note 4. To change the frequency of the PLL selected as the system clock source, it is necessary to change the system clock source to another clock source, stop the PLL, and then change the PLL setting.

When the PLL1 is selected as the system clock source, additional procedure is required for procedure No.7. This additional procedure is shown in [Figure 8.14](#). It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt



**Figure 8.14** Additional flow of system clock setting when PLL is used for the system clock source and CPUCLK to be faster frequency

### 8.11.1.3 System Clock Setting for Slower frequency

An example of setting procedure when changing the current system clock frequency to a slower frequency is shown in [Table 8.8](#).

**Table 8.8 Example of Setting Procedure When changing the current system clock frequency to a Slower frequency**

No.	Step	Description
1	Start	
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 <sup>*1*2</sup> bit in PRCR register
3	Change Operating Power Control Mode to High-Speed mode <sup>*1</sup>	Change to High-Speed mode with OPCCR register. <sup>*1</sup> For details, see <a href="#">section 10, Low Power Modes</a> .
4	Set a clock source to oscillate <sup>*3</sup>	Set a clock source that switches to the system clock source to oscillate. <sup>*4</sup>
5	Set internal clock division ratio	Make the following settings so that the frequency does not exceed the electrical characteristics range when the system clock source is switched in the next step. <ul style="list-style-type: none"> <li>• Set the internal clock division ratio larger than the current division ratio.</li> <li>• Set the internal clock division ratio so that the frequency after switching the system clock source is slower than the current frequency.<sup>*3</sup></li> </ul>
6	Switch the System Clock source <sup>*3</sup>	Switch the system clock source with SCKSCR register. Before switching the system clock source, confirm that the clock source oscillation is stable.
7	Re-set internal clock division ratio	Re-set the internal clock division ratio with SCKDIVCR register according to the frequency after switching the system clock source.
8	Set Flash Wait Cycle	Set an appropriate access wait count for the flash memory according to the frequency at which it operates after the system clock (ICLK) is set. For details, see <a href="#">section 49, Flash Memory</a> .
9	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. <sup>*2</sup>
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 <sup>*1*2</sup> bit in PRCR register
11	End	System clock is changed to a Slower frequency.

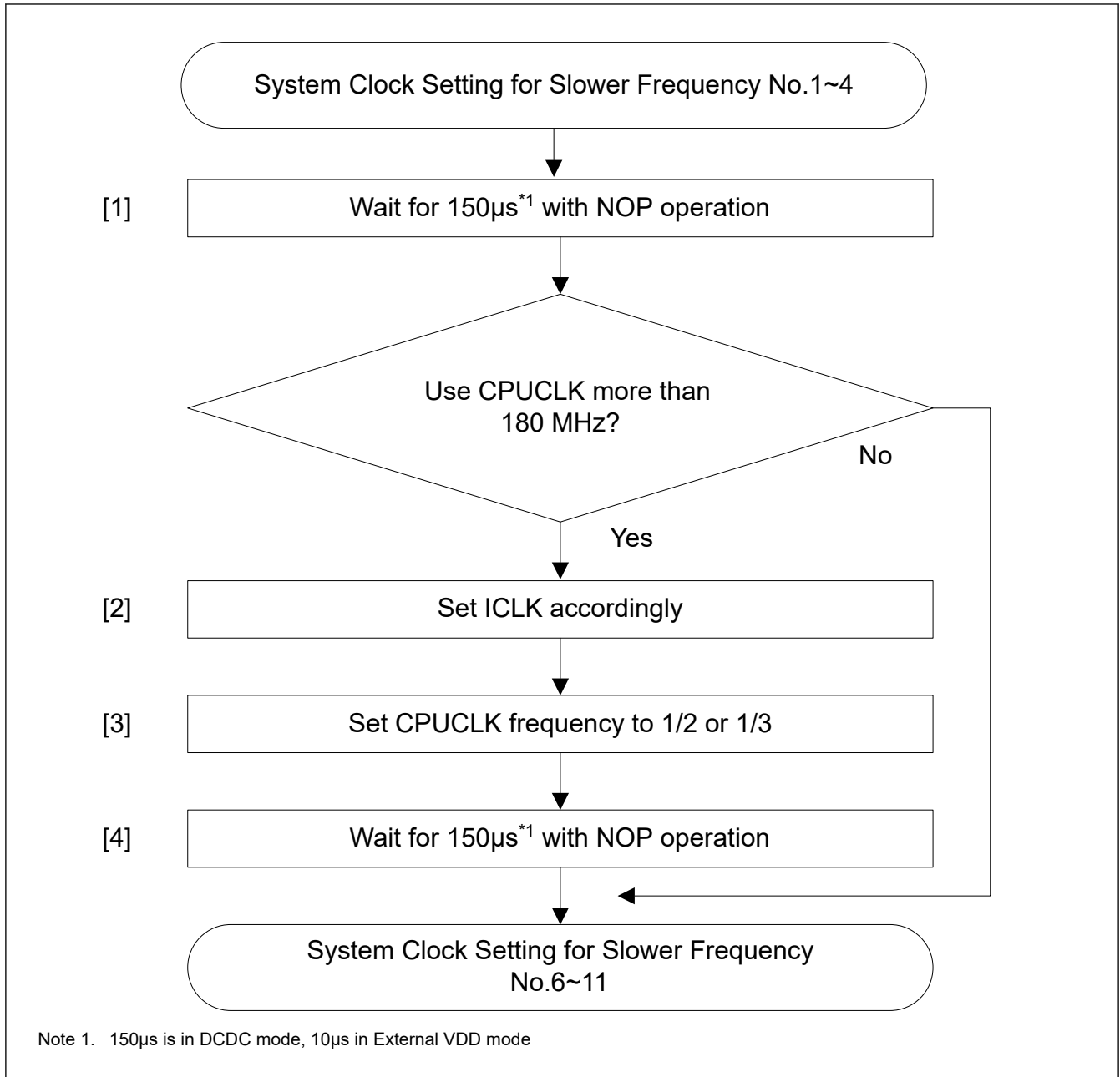
Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.

Note 2. Setting is not necessary if the operating power control mode is not changed.

Note 3. Setting is not necessary if the system clock source is not changed.

Note 4. To change the frequency of the PLL selected as the system clock source, it is necessary to change the system clock source to another clock source, stop the PLL, and then change the PLL setting.

When the PLL1 is selected as the system clock source, additional procedure is required for procedure No.5. This additional procedure is shown in [Figure 8.15](#). It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt.



**Figure 8.15 Additional flow of system clock setting when PLL is used for the system clock source and CPUCLK to be slower frequency**

### 8.11.2 Peripheral module-dedicated Clock Setting

This section describes the setting procedure of the operating clock for the peripheral module. The target clocks are as follows : SCICLK, SPICLK, OCTACLK, CANFDCLK, USBCLK.

#### 8.11.2.1 Initial Peripheral module-dedicated Clock Setting

An example of the initial clock setting procedure for peripheral module is shown in [Table 8.9](#).

**Table 8.9 Example of the setting procedure of the operating clock for peripheral module after reset release / after Deep Software Standby cancellation (1 of 2)**

No.	Step	Description
1	Start	After reset release / Deep software standby cancellation (MOCO is selected as system clock source)

**Table 8.9 Example of the setting procedure of the operating clock for peripheral module after reset release / after Deep Software Standby cancellation (2 of 2)**

No.	Step	Description
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 <sup>*1</sup> *2 bit in PRCR register
3	Change Operating Power Control Mode to High-Speed mode <sup>*1</sup>	Change to High-Speed mode with OPCCR register. <sup>*1</sup> For details, see <a href="#">section 10, Low Power Modes</a> .
4	Set CKSREQ and wait for setting CKSRDY	Write 1 to CKSREQ bit in each CKCR register. Polling until CKSRDY bit in each CKCR register is read as 1 While CKSRDY is 1, the clock supplying stops.
5	Set a clock source to oscillate	Set a clock source that switches to the peripheral module-dedicated clock source to oscillate.
6	Set the clock division ratio and switch the clock source	Write the setting value to CKDIV[2:0] in each CKDIVCR register. Write the setting value to CKSEL[3:0] in each CKCR register.
7	Clear CKSREQ and wait for clearing CKSRDY	Write 0 to CKSREQ bit in each CKCR register. Polling until CKSRDY bit in each CKCR register is read as 0. After CKSRDY is 0, the clock supplying starts.
8	Cancel the Module Stop Control	Cancel the Module Stop Control in MSTPCRn register (n = A to E) and supply clock to peripheral module.
9	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. <sup>*2</sup>
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
11	End	Clock setting is completed. And then, set the peripheral modules.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.

Note 2. Setting is not necessary if the operating power control mode is not changed.

### 8.11.2.2 Peripheral module-dedicated Clock source changing

[Table 8.10](#) shows an example of the setting change procedure when switching the clock source of the operating clock for peripheral module in use.

**Table 8.10 Example of the setting change procedure when switching the clock source of the operating clock for peripheral module in use (1 of 2)**

No.	Step	Description
1	Start	The operating clock for the peripheral module is being used.
2	Stop the peripheral module	Stop the operation of the peripheral module that the operating clock is to be changed.
3	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 <sup>*1</sup> *2 bit in PRCR register
4	Change Operating Power Control Mode to High-Speed mode <sup>*1</sup>	Change to High-Speed mode with OPCCR register. <sup>*1</sup> For details, see <a href="#">section 10, Low Power Modes</a> .
5	Set CKSREQ and wait for setting CKSRDY	Write 1 to CKSREQ bit in each CKCR register. Polling until CKSRDY bit in each CKCR register is read as 1. While CKSRDY is 1, the clock supplying stops.
6	Set a clock source to oscillate	Set a clock source that switches to the peripheral module-dedicated clock source to oscillate.
7	Set the clock division ratio and switch the clock source	Write the setting value to CKDIV[2:0] in each CKDIVCR register. Write the setting value to CKSEL[3:0] in each CKCR register.
8	Clear CKSREQ and wait for clearing CKSRDY	Write 0 to CKSREQ bit in each CKCR register. Polling until CKSRDY bit in each CKCR register is read as 0. After CKSRDY is 0, the clock supplying starts.
9	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. <sup>*2</sup>
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register



**Table 8.10 Example of the setting change procedure when switching the clock source of the operating clock for peripheral module in use (2 of 2)**

No.	Step	Description
11	Restart the peripheral module	Change the peripheral module settings according to the operating clock after the settings are changed. After that, restart the operation of the peripheral module.
12	End	Clock setting is completed.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.

Note 2. Setting is not necessary if the operating power control mode is not changed.

### 8.11.2.3 Peripheral module-dedicated Clock division ratio changing

Table 8.11 shows an example of the setting change procedure when switching the switching the division ratio of the operating clock for peripheral module in use.

**Table 8.11 Example of the setting change procedure when switching the division ratio of the operating clock for peripheral module in use (without the clock source change)**

No.	Step	Description
1	Start	The operating clock for the peripheral module is being used.
2	Stop the peripheral module	Stop the operation of the peripheral module that the operating clock is to be changed.
3	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 <sup>*1</sup> *2 bit in PRCR register
4	Change Operating Power Control Mode to High-Speed mode <sup>*1</sup>	Change to High-Speed mode with OPCCR register. <sup>*1</sup> For details, see <a href="#">section 10, Low Power Modes</a> .
5	Set the clock division ratio	Write the setting value to CKDIV[2:0] in each CKDIVCR register.
6	Change Operating Power Control Mode	Change the Operating Power Control Mode with OPCCR register. <sup>*2</sup>
7	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
8	Restart the peripheral module	Change the peripheral module settings according to the operating clock after the settings are changed. After that, restart the operation of the peripheral module.
9	End	Clock setting is completed.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.

Note 2. Setting is not necessary if the operating power control mode is not changed.

### 8.11.3 Main Clock Oscillator Setting

Table 8.12 shows an example of the initial setting procedure for the Main Clock Oscillator.

**Table 8.12 Example of the initial setting procedure for Main Clock Oscillator after reset release / after Deep Software Standby cancellation (1 of 2)**

No.	Step	Description
1	Start	The Main Clock Oscillator is stopped after reset release / Deep software standby cancellation
2	Cancel Register Write Protection	Set 1 to PRC0 bit in PRCR register
3	Set the mode of Main Clock Oscillator	Set the following Main Clock Oscillator mode with MOMCR register <ul style="list-style-type: none"> <li>• Resonator or Clock input</li> <li>• Auto Gain Control function</li> <li>• Drive Capability</li> </ul>
4	Set the oscillation stabilization wait time	Set the Main Clock Oscillator Wait Time in MOSCWTCR register.
5	Set the oscillation keep in Software Standby mode	Set MOSCSCR register if Main Clock Oscillator keeps oscillation in Software Standby mode.
6	Set the Main Clock Oscillator to operate	Set the Main Clock Oscillator to start oscillating with the MOSCCR register.

**Table 8.12 Example of the initial setting procedure for Main Clock Oscillator after reset release / after Deep Software Standby cancellation (2 of 2)**

No.	Step	Description
7	Wait for Main Clock oscillation to stabilize	Polling until MOSCSF bit in OSCSF register is read as 1 (Oscillation is stable)
8	Apply Register Write Protection	Set 0 to PRC0 bit in PRCR register
9	End	The Main Clock Oscillator setting is completed. The Main Clock is available.

### 8.11.4 Sub-Clock Oscillator Setting

Table 8.13 shows an example of the initial setting procedure for the Sub-Clock Oscillator.

**Table 8.13 Example of the initial setting procedure for Sub-Clock Oscillator after VBAT Power On Reset release**

No.	Step	Description
1	Start	The Sub-Clock Oscillator is stopped after VBAT Power On Reset release.
2	Cancel Register Write Protection	Set 1 to PRC0 bit in PRCR register
3	Setting the mode of Sub-Clock Oscillator	Set the following Sub-Clock Oscillator mode with SOMCR register <ul style="list-style-type: none"> <li>Resonator or Clock input</li> <li>Drive Capability</li> </ul>
4	Set the Sub-Clock Oscillator to operate	Set the Sub-Clock Oscillator to start oscillating with the SOSCCR register.
5	Wait for Sub-Clock oscillation to stabilize	Wait for sub-clock oscillation stabilization time (tSUBOSCWT). For the sub-clock oscillation stabilization time, see <a href="#">section 52, Electrical Characteristics</a> .
6	Apply Register Write Protection	Set 0 to PRC0 bit in PRCR register
7	End	The Sub-Clock Oscillator setting is completed. The Sub-Clock is available.

### 8.11.5 High-speed On Chip Oscillator Setting

#### 8.11.5.1 HOCO Initial Clock Setting

Table 8.14 shows an example of the initial setting procedure for the HOCO.

**Table 8.14 Example of the HOCO initial setting procedure after reset release / after Deep Software Standby cancellation (OFS1(\_SEC).HOCOEN = 1, without FLL)**

No.	Step	Description
1	Start	The HOCO is stopped after reset release / Deep software standby cancellation when OFS1(_SEC).HOCOEN is 1.
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register
3	Check the HOCO power supply*1	Check the following bit in the HOCOLDOCR register <ul style="list-style-type: none"> <li>LDOSTP bit is 0 (LDO is enabled)</li> </ul>
4	Set the HOCO frequency	Set the HOCO frequency with HOCOCCR2 register
5	Set the oscillation keep in Software Standby mode	If HOCO keeps oscillation in Software Standby mode, set the following: <ul style="list-style-type: none"> <li>HOCOSOKP bit in HOCOSCR register</li> <li>SKEEP bit in HOCOLDOCR register</li> </ul>
6	Set HOCO to operate	Set HOCO to start oscillating with the HOCOCCR register.
7	Wait for HOCO clock oscillation to stabilize	Polling until HOCOSF bit in OSCSF register is read as 1 (Oscillation is stable)
8	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
9	End	HOCO clock setting is completed.HOCO Clock is available.

Note 1. When HOCOLDOCR.LDOSTP bit is 1, do not operate HOCO. If HOCOLDOCR.LDOSTP is switched from 1 to 0, waiting for the HOCO power supply to stabilize is required. See [section 52, Electrical Characteristics](#) on waiting time for the HOCO power supply to stabilize.

### 8.11.5.2 HOCO Setting with FLL function

Table 8.15 shows an example of the initial setting procedure for HOCO with FLL function.

Table 8.16 shows the FLL function setting flow for the Software Standby mode transition and cancellation.

**Table 8.15 Example of the HOCO setting procedure with FLL function after reset release / after Deep Software Standby cancellation (OFS1(\_SEC).HOCOEN = 1)**

No.	Step	Description
1	Start	The HOCO is stopped after reset release / Deep software standby cancellation when OFS1(_SEC).HOCOEN is 1.
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register
3	Check the HOCO power supply <sup>*1</sup>	Check the following bit in the HOCOLDOCR register. <ul style="list-style-type: none"> <li>LDOSTP bit is 0 (LDO is enabled)</li> </ul>
4	Set the HOCO frequency	Set the HOCO frequency with HOCOCCR2 register
5	Set the FLL function to enable <sup>*2</sup>	Set the FLL Multiplication Control with FLLCR2 register. Set the FLL function to enable with FLLCR1 register.
6	Set HOCO to operate	Set HOCO to start oscillating with the HOCOCCR register.
7	Wait for HOCO clock oscillation to stabilize	Polling until HOCOSF bit in OSCSF register is read as 1 (Oscillation is stable)
8	Wait for FLL stabilization	Wait for FLL stabilization wait time ( $t_{FLLWT}$ ), or wait until the HOCO clock is measured to confirm that the frequency accuracy is stable.
9	Check the HOCO stabilization	Check that HOCOSF bit in OSCSF register is read as 1
10	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
11	End	FLL setting is completed. HOCO Clock is available.

Note 1. When HOCOLDOCR.LDOSTP bit is 1, do not operate HOCO.

If HOCOLDOCR.LDOSTP is switched from 1 to 0, waiting for the HOCO power supply to stabilize is required. See [section 52, Electrical Characteristics](#) on waiting time for the HOCO power supply to stabilize.

Note 2. Sub-Clock Oscillator must be oscillating with the stabilization.

**Table 8.16 FLL setting flow for Software Standby mode transition and cancellation**

No.	Step	Description
1	Start	The HOCO is oscillating with FLL is enabled
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register
3	Stop HOCO <sup>*1</sup>	Set HOCO to stop with HOCOCCR register.
4	Disable FLL	Set the FLL function to disable with FLLCR1 register.
5	WFI instruction <sup>*2</sup>	Transition to Software Standby mode with WFI instruction
6	Software Standby mode <sup>*2</sup>	MCU is in Software Standby mode
7	Cancellation Software Standby mode <sup>*2</sup>	Software Standby mode is cancelled
8	Set the FLL function to enable <sup>*3</sup>	Set the FLL Multiplication Control with FLLCR2 register. Set the FLL function to enable with FLLCR1 register.
9	Set HOCO to operate	Set HOCO to start oscillating with the HOCOCCR register.
10	Wait for FLL stabilization	Wait for FLL stabilization wait time ( $t_{FLLWT}$ ), or wait until the HOCO clock is measured to confirm that the frequency accuracy is stable.
11	Check the HOCO stabilization	Check that HOCOSF bit in OSCSF register is read as 1.
12	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
13	End	HOCO Clock is available.

Note 1. If HOCO is used as the system clock source or the clock source of PLL, these clock source must be changed to another clock before HOCO is stopped.

Note 2. See [section 10, Low Power Modes](#) for detail of Software Standby mode.

Note 3. Sub-Clock Oscillator must be oscillating with the stabilization.

## 8.11.6 PLL Setting

### 8.11.6.1 PLL Initial Clock Setting

Table 8.17 shows an example of the initial setting procedure for the PLL.

**Table 8.17 Example of the PLL initial setting procedure after reset release / after Deep Software Standby cancellation (PLL1 or PLL2)**

No.	Step	Description
1	Start	The PLL1 is stopped after reset release / Deep software standby cancellation. The PLL2 is stopped after reset release / Deep software standby cancellation.
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register.
3	Change Operating Power Control Mode to High-Speed mode <sup>*1</sup>	Change to High-Speed mode with OPCCR register. For details, see <a href="#">section 10, Low Power Modes</a> .
4	Check the PLL power supply <sup>*2</sup>	[PLL1] Check the following bit in the PLL1LDOCR register. <ul style="list-style-type: none"> <li>LDOSTP bit is 0 (LDO is enabled).</li> </ul> [PLL2] Check the following bit in the PLL2LDOCR register. <ul style="list-style-type: none"> <li>LDOSTP bit is 0 (LDO is enabled).</li> </ul>
5	Configure the PLL settings	[PLL1] Configure the following PLL1 settings in PLLCCR and PLLCCR2 registers. [PLL2] Configure the following PLL2 settings in PLL2CCR and PLL2CCR2 registers. [Setting Items] <ul style="list-style-type: none"> <li>PLL input frequency division ratio</li> <li>PLL clock source</li> <li>Frequency Multiplication Factor</li> <li>Frequency Multiplication Fractional Factor</li> <li>PLL Output Frequency Division Ratio (P/Q/R)</li> </ul>
6	Set PLL to operate	[PLL1] Set PLL1 to start oscillating with the PLLCR register. [PLL2] Set PLL2 to start oscillating with the PLL2CR register.
7	Wait for PLL clock oscillation to stabilize	[PLL1] Polling until PLLSF bit in OSCSF register is read as 1 (Oscillation is stable). [PLL2] Polling until PLL2SF bit in OSCSF register is read as 1 (Oscillation is stable).
8	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register
9	End	PLL1 or PLL2 clock setting is completed. [PLL1] PLL1P, PLL1Q, PLL1R clocks are available. [PLL2] PLL2P, PLL2Q, PLL2R clocks are available.

Note 1. Setting is not necessary if the operating power control mode is already the High-Speed mode.

Note 2. When PLL1LDOCR.LDOSTP bit is 1, do not operate PLL1.

If PLL1LDOCR.LDOSTP is switched from 1 to 0, waiting for the PLL1 power supply to stabilize is required. See [section 52, Electrical Characteristics](#) on waiting time for the PLL1 power supply to stabilize.

When PLL2LDOCR.LDOSTP bit is 1, do not operate PLL2.

If PLL2LDOCR.LDOSTP is switched from 1 to 0, waiting for the PLL2 power supply to stabilize is required. See [section 52, Electrical Characteristics](#) on waiting time for the PLL2 power supply to stabilize.

### 8.11.6.2 PLL Clock Setting Change

Table 8.18 shows an example of the initial setting procedure for the PLL.

**Table 8.18 Example of the PLL setting change procedure (PLL1 or PLL2)**

No.	Step	Description
1	Start	PLL1 is oscillating and the PLL1 clocks (PLL1P, PLL1Q, PLL1R) are being used. PLL2 is oscillating and the PLL2 clocks (PLL2P, PLL2Q, PLL2R) are being used.
2	Cancel Register Write Protection	Set 1 to PRC0 bit and PRC1 bit in PRCR register.
3	Stop PLL *1	[PLL1] Set PLL1 to stop with PLLCR register. [PLL2] Set PLL2 to stop with PLL2CR register.
4	Change the PLL settings	[PLL1] Configure the following PLL1 settings in PLLCCR and PLLCCR2 registers. [PLL2] Configure the following PLL2 settings in PLL2CCR and PLL2CCR2 registers. [Setting Items] <ul style="list-style-type: none"> <li>• PLL input frequency division ratio</li> <li>• PLL clock source</li> <li>• Frequency Multiplication Factor</li> <li>• Frequency Multiplication Fractional Factor</li> <li>• PLL Output Frequency Division Ratio (P/Q/R)</li> </ul>
5	Set PLL to operate	[PLL1] Set PLL1 to start oscillating with the PLLCR register. [PLL2] Set PLL2 to start oscillating with the PLL2CR register.
6	Wait for PLL clock oscillation to stabilize	[PLL1] Polling until PLLSF bit in OSCSF register is read as 1 (Oscillation is stable). [PLL2] Polling until PLL2SF bit in OSCSF register is read as 1 (Oscillation is stable).
7	Apply Register Write Protection	Set 0 to PRC0 bit and PRC1 bit in PRCR register.
8	End	PLL1 or PLL2 clock setting change is completed. [PLL1] PLL1P, PLL1Q, PLL1R clocks are available. [PLL2] PLL2P, PLL2Q, PLL2R clocks are available.

Note 1. If PLL1P clock is used as the system clock source, the system clock source must be changed to another clock source before PLL1 is stopped.

## 8.12 Usage Notes

### 8.12.1 Notes on Clock Generation Circuit

The frequency of the following clocks supplied to each module changes according to the setting of the SCKDIVCR and SCKDIVCR2 register:

- CPU clock (CPUCLK)
- System clock (ICLK)
- Peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD and PCLKE)
- FlashIF clock (FCLK)

Each frequency must meet the following conditions:

- Each frequency must be selected within the operation-guaranteed range of the operating frequency (f) specified in the AC characteristics. See [section 52, Electrical Characteristics](#).
- Each clock must be set according to [Table 8.2](#).

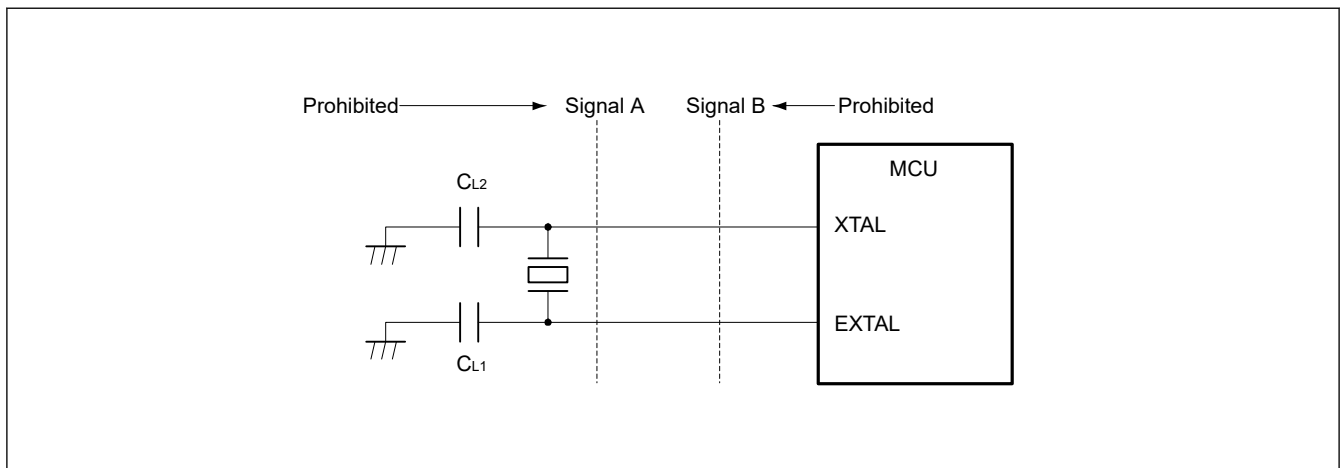
To ensure correct processing after the clock frequency changes, first write to the relevant Clock Control register to change the frequency, then read the value from the register, and finally perform the subsequent processing.

### 8.12.2 Notes on Resonator

Because various resonator characteristics relate closely to your board design, adequate evaluation is required before use. See the resonator connection example in [Figure 8.5](#). The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Therefore, consult the resonator manufacturer when determining the circuit constants. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

### 8.12.3 Notes on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in [Figure 8.16](#) to prevent electromagnetic induction from interfering with correct oscillation. [Figure 8.16](#) shows the case which the main clock oscillator is used. In case of sub-clock oscillator, it is also the same as [Figure 8.16](#).



**Figure 8.16** Signal routing in board design for oscillation circuit

### 8.12.4 Notes on Resonator Connect Pin

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports. When these pins are used as general ports, the main clock must be stopped (MOSCCR.MOSTP bit should be set to 1).

### 8.12.5 Notes on Using Sub-Clock Oscillator

The output of the P212 (EXTAL), P213 (XTAL) and P706 pins may affect the oscillation by the sub-clock oscillator.

If the sub-clock oscillator is used, implement board design so as not to affect the oscillation. Renesas strongly recommends setting the PmnPFS.DSCR[1:0] bits to 00b or 01b when using the P212 (EXTAL), P213 (XTAL) and P706 as output pins and using the sub-clock oscillator.

In addition, when using the sub-clock oscillator in Low power mode drive capability (SOMCR.SODRV[1:0]= 01b, 10b and 11b), Renesas recommends setting the PmnPFS.DSCR[1:0] bits to 00b when using the P212 (EXTAL), P213 (XTAL) and P706 as output pins and using the sub-clock oscillator.

## 9. Clock Frequency Accuracy Measurement Circuit (CAC)

### 9.1 Overview

The Clock Frequency Accuracy Measurement Circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock selected as the measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range. When measurement is complete or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.

[Table 9.1](#) lists the CAC specifications, [Figure 9.1](#) shows the CAC block diagram, and [Table 9.2](#) lists the CAC I/O pin.

**Table 9.1 CAC specifications**

Parameter	Specifications
Measurement target clocks	Frequency can be measured for: <ul style="list-style-type: none"> <li>• Main clock oscillator (CACMCLK)</li> <li>• Sub-clock oscillator (CACCLK)</li> <li>• HOCO clock (CACHCLK)</li> <li>• MOCO clock (CACMOCLK)</li> <li>• LOCO clock (CACLCLK)</li> <li>• Peripheral module clock B (PCLKB)</li> </ul>
Measurement reference clocks	Frequency can be referenced to: <ul style="list-style-type: none"> <li>• External clock input to the CACREF pin</li> <li>• Main clock oscillator (CACMCLK)</li> <li>• Sub-clock oscillator (CACCLK)</li> <li>• HOCO clock (CACHCLK)</li> <li>• MOCO clock (CACMOCLK)</li> <li>• LOCO clock (CACLCLK)</li> <li>• Peripheral module clock B (PCLKB)</li> </ul>
Selectable function	Digital filter
Interrupt sources	<ul style="list-style-type: none"> <li>• Measurement end</li> <li>• Frequency error</li> <li>• Overflow</li> </ul>
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set

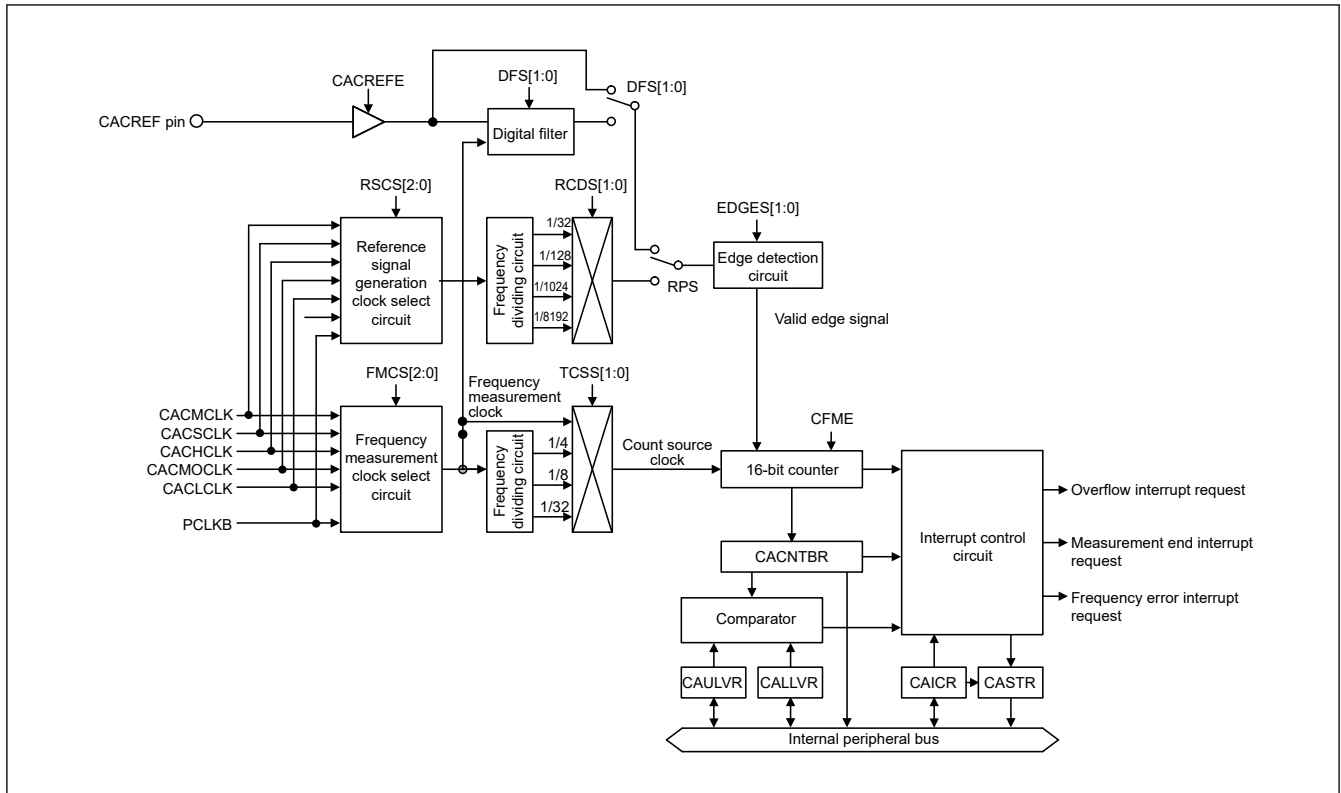


Figure 9.1 CAC block diagram

Table 9.2 CAC I/O pin

Function	Pin name	I/O	Description
CAC	CACREF	Input	Measurement reference clock input pin

## 9.2 Register Descriptions

### 9.2.1 CACR0 : CAC Control Register 0

Base address: CAC = 0x4020\_2400  
CAC\_NS = 0x5020\_2400

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFME

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	CFME	Clock Frequency Measurement Enable 0: Disable 1: Enable	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

#### CFME bit (Clock Frequency Measurement Enable)

The CFME bit enables clock frequency measurement. Changes made to this bit are not immediately reflected to the internal circuit. Read the bit to confirm that the change has been reflected.



### 9.2.2 CACR1 : CAC Control Register 1

Base address: CAC = 0x4020\_2400  
CAC\_NS = 0x5020\_2400

Offset address: 0x01

Bit position: 7 6 5 4 3 2 1 0

Bit field:	EDGES[1:0]	TCSS[1:0]	FMCS[2:0]	CACR EFE
------------	------------	-----------	-----------	-------------

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	CACREFE	CACREF Pin Input Enable 0: Disable 1: Enable	R/W
3:1	FMCS[2:0]	Measurement Target Clock Select 0 0 0: Main clock oscillator (CACMCLK) 0 0 1: Sub-clock oscillator (CACSCCLK) 0 1 0: HOCO clock (CACHCLK) 0 1 1: MOCO clock (CACMOCLK) 1 0 0: LOCO clock (CACLCLK) 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
5:4	TCSS[1:0]	Timer Count Clock Source Select 0 0: No division 0 1: × 1/4 clock 1 0: × 1/8 clock 1 1: × 1/32 clock	R/W
7:6	EDGES[1:0]	Valid Edge Select 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note: S-TYPE3, P-TYPE3

Note: Set the CACR1 register when the CACR0.CFME bit is 0.

#### CACREFE bit (CACREF Pin Input Enable)

The CACREFE bit enables the CACREF pin input.

#### FMCS[2:0] bits (Measurement Target Clock Select)

The FMCS[2:0] bits select the measurement target clock whose frequency is to be measured.

#### TCSS[1:0] bits (Timer Count Clock Source Select)

The TCSS[1:0] bits select the division ratio of the measurement target clock.

#### EDGES[1:0] bits (Valid Edge Select)

The EDGES[1:0] bits select the valid edge for the reference signal.

### 9.2.3 CACR2 : CAC Control Register 2

Base address: CAC = 0x4020\_2400  
CAC\_NS = 0x5020\_2400

Offset address: 0x02

Bit position: 7 6 5 4 3 2 1 0

Bit field:	DFS[1:0]	RCDS[1:0]	RSCS[2:0]	RPS
------------	----------	-----------	-----------	-----

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	RPS	Reference Signal Select 0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
3:1	RSCS[2:0]	Measurement Reference Clock Select 0 0 0: Main clock oscillator (CACMCLK) 0 0 1: Sub-clock oscillator (CACSCCLK) 0 1 0: HOCO clock (CACHCLK) 0 1 1: MOCO clock (CACMOCLK) 1 0 0: LOCO clock (CACLCLK) 1 0 1: Peripheral module clock B (PCLKB) 1 1 0: Setting prohibited 1 1 1: Setting prohibited	R/W
5:4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ratio Select 0 0: × 1/32 clock 0 1: × 1/128 clock 1 0: × 1/1024 clock 1 1: × 1/8192 clock	R/W
7:6	DFS[1:0]	Digital Filter Select 0 0: Disable digital filtering 0 1: Use sampling clock for the digital filter as the frequency measuring clock 1 0: Use sampling clock for the digital filter as the frequency measuring clock divided by 4 1 1: Use sampling clock for the digital filter as the frequency measuring clock divided by 16.	R/W

Note: S-TYPE3, P-TYPE3

Note: Set the CACR2 register when the CACR0.CFME bit is 0.

#### RPS bit (Reference Signal Select)

The RPS bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

#### RSCS[2:0] bits (Measurement Reference Clock Select)

The RSCS[2:0] bits select the reference clock for measurement.

#### RCDS[1:0] bits (Measurement Reference Clock Frequency Division Ratio Select)

The RCDS[1:0] bits select the frequency-divisor of the reference clock for measurement when an internal reference clock is selected. When RPS = 0 (CACREF pin is used as the reference clock source), the reference clock is not divided.

#### DFS[1:0] bits (Digital Filter Select)

The DFS[1:0] bits enable or disable the digital filter and selects its sampling clock.

### 9.2.4 CAICR : CAC Interrupt Control Register

Base address: CAC = 0x4020\_2400  
CAC\_NS = 0x5020\_2400

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	OVFF CL	MEND FCL	FERR FCL	—	OVFIE	MEND IE	FERRI E

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	FERRIE	Frequency Error Interrupt Request Enable 0: Disable 1: Enable	R/W
1	MENDIE	Measurement End Interrupt Request Enable 0: Disable 1: Enable	R/W
2	OVFIE	Overflow Interrupt Request Enable 0: Disable 1: Enable	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	FERRFCL	FERRF Clear 0: No effect 1: The CASTR.FERRF flag is cleared	W
5	MENDFCL	MENDF Clear 0: No effect 1: The CASTR.MENDF flag is cleared	W
6	OVFFCL	OVFF Clear 0: No effect 1: The CASTR.OVFF flag is cleared.	W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

#### **FERRIE bit (Frequency Error Interrupt Request Enable)**

The FERRIE bit enables or disables the frequency error interrupt request.

#### **MENDIE bit (Measurement End Interrupt Request Enable)**

The MENDIE bit enables or disables the measurement end interrupt request.

#### **OVFIE bit (Overflow Interrupt Request Enable)**

The OVFIE bit enables or disables the overflow interrupt request.

#### **FERRFCL bit (FERRF Clear)**

Setting the FERRFCL bit to 1 clears the CASTR.FERRF flag.

#### **MENDFCL bit (MENDF Clear)**

Setting the MENDFCL bit to 1 clears the CASTR.MENDF flag.

#### **OVFFCL bit (OVFF Clear)**

Setting the OVFFCL bit to 1 clears the CASTR.OVFF flag.

### 9.2.5 CASTR : CAC Status Register

Base address: CAC = 0x4020\_2400  
CAC\_NS = 0x5020\_2400

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	OVFF	MEND F	FERR F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FERRF	Frequency Error Flag 0: Clock frequency is within the allowable range 1: Clock frequency has deviated beyond the allowable range (frequency error).	R
1	MENDF	Measurement End Flag 0: Measurement is in progress 1: Measurement ended	R
2	OVFF	Overflow Flag 0: Counter has not overflowed 1: Counter overflowed	R
7:3	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

#### FERRF flag (Frequency Error Flag)

The FERRF flag indicates a deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside the allowable range defined in the CAULVR and CALLVR registers.

[Clearing condition]

- 1 is written to the FERRFCL bit.

#### MENDF flag (Measurement End Flag)

The MENDF flag indicates the end of measurement.

[Setting condition]

- Measurement ends.

[Clearing condition]

- 1 is written to the MENDFCL bit.

#### OVFF flag (Overflow Flag)

The OVFF flag indicates that the counter overflowed.

[Setting condition]

- The counter overflows.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

### 9.2.6 CAULVR : CAC Upper-Limit Value Setting Register

Base address: CAC = 0x4020\_2400  
CAC\_NS = 0x5020\_2400

Offset address: 0x06

Bit position: 15 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	n/a	The Upper Value of the Allowable Range The CAULVR register is a 16-bit read/write register that specifies the upper value of the allowable range. When the counter value exceeds the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0. The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit, and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.	R/W

Note: S-TYPE3, P-TYPE3

### 9.2.7 CALLVR : CAC Lower-Limit Value Setting Register

Base address: CAC = 0x4020\_2400  
CAC\_NS = 0x5020\_2400

Offset address: 0x08

Bit position: 15 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	n/a	The Lower Value of the Allowable Range The CALLVR register is a 16-bit read/write register that specifies the lower value of the allowable range. When the counter value falls below the value specified in this register, a frequency error is detected. Write to this register when the CACR0.CFME bit is 0. The counter value stored in CACNTBR can vary depending on the difference between the phases of the digital filter and edge-detection circuit, and the signal on the CACREF pin. Ensure that this setting allows an adequate margin.	R/W

Note: S-TYPE3, P-TYPE3

### 9.2.8 CACNTBR : CAC Counter Buffer Register

Base address: CAC = 0x4020\_2400  
CAC\_NS = 0x5020\_2400

Offset address: 0x0A

Bit position: 15 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

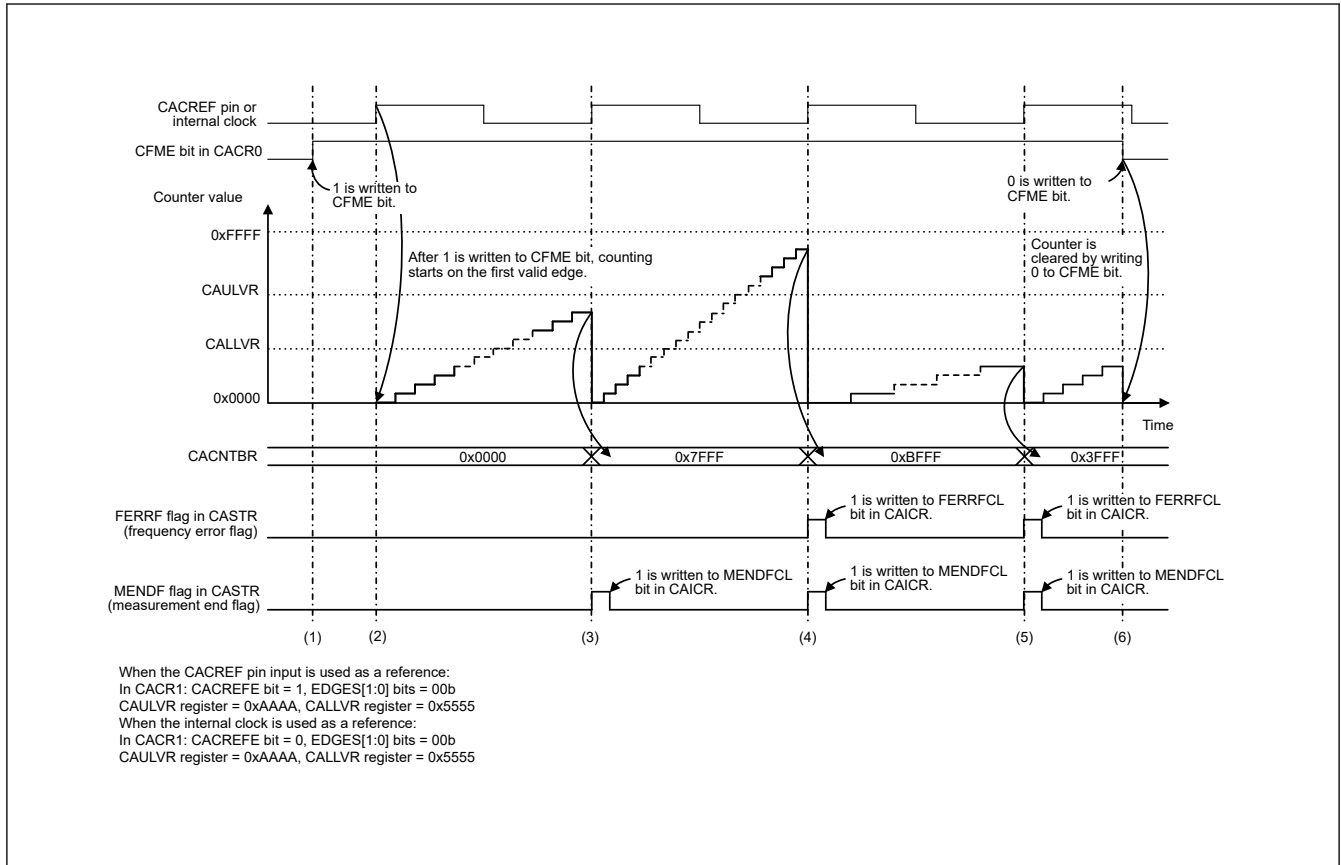
Bit	Symbol	Function	R/W
15:0	n/a	The Measurement Result The CACNTBR register is a 16-bit read-only register that stores the measurement result.	R

Note: S-TYPE3, P-TYPE3

## 9.3 Operation

### 9.3.1 Measuring Clock Frequency

The CAC measures the clock frequency using the CACREF pin input or an internal clock as a reference. Figure 9.2 shows an operating example of the CAC.



**Figure 9.2 CAC operating example**

The events in Figure 9.2 are:

1. When the CACREF pin input is used as reference (CACR1.CACREFE = 1), frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is set to 0 and the CACR1.CACREFE bit is set to 1. When the internal clock is used as reference (CACR1.CACREFE = 0), frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is set to 1.
2. When the CACREF pin input is used as reference, after 1 is written to the CFME bit, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits (rising edge (CACR1.EDGES[1:0] = 00b) in Figure 9.2) is input from the CACREF pin. When the internal clock is used as reference, after 1 is written to the CFME bit, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits (rising edge (CACR1.EDGES[1:0] = 00b) in Figure 9.2) is input based on the clock source selected by the CACR2.RSCS[2:0] bits.
3. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If both  $CACNTBR \leq CAULVR$  and  $CACNTBR \geq CALLVR$  are true, only the MENDF flag in CASTR is set to 1, because the clock frequency is correct. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
4. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If  $CACNTBR > CAULVR$ , the FERRF flag in CASTR is set to 1, because the clock frequency is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.
5. When the next valid edge is input, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR. If  $CACNTBR < CALLVR$ , the FERRF flag in CASTR is set to 1, because the clock frequency

is erroneous. If the FERRIE bit in CAICR is 1, a frequency error interrupt is generated. The MENDF flag in CASTR is set to 1 at the end of measurement. If the MENDIE bit in CAICR is 1, a measurement end interrupt is generated.

- When the CFME bit in CACR0 is 1, the counter value is transferred to CACNTBR and compared with the values in CAULVR and CALLVR every time a valid edge is input. Writing 0 to the CFME bit in CACR0 clears the counter and stops up-counting.

### 9.3.2 Digital Filtering of Signals on CACREF Pin

The CACREF pin has a digital filter, and levels on the CACREF pin are transmitted to the internal circuitry after three consecutive matches in the selected sampling interval. The same level continues to be transmitted internally until the level on the pin has three consecutive matches again. Enabling or disabling of the digital filter and its sampling clock are selectable.

The counter value transferred to CACNTBR might be in error by up to 1 cycle of the sampling clock because of the difference between the phases of the digital filter and the signal input to the CACREF pin. When a frequency dividing clock is selected as a count source clock, the counter value error is obtained using the following formula:

$$\text{Counter value error} = (1 \text{ cycle of the count source clock}) / (1 \text{ cycle of the sampling clock})$$

## 9.4 Interrupt Requests

The CAC generates three types of interrupt requests:

- Frequency error interrupt
- Measurement end interrupt
- Overflow interrupt

When an interrupt source is generated, the associated status flag is set to 1. [Table 9.3](#) provides information on the CAC interrupt requests.

**Table 9.3 CAC interrupt requests**

Interrupt request	Interrupt enable bit	Status flag	Interrupt sources
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR with CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	<ul style="list-style-type: none"> <li>• Valid edge is input from the CACREF pin or internal clock</li> <li>• Measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit</li> </ul>
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	Counter overflows

## 9.5 Usage Notes

### 9.5.1 Settings for the Module-Stop Function

The Module Stop Control Register C (MSTPCRC) can enable or disable CAC operation. The CAC module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

## 10. Low Power Modes

### 10.1 Overview

The MCU has several functions for reducing power consumption, such as setting clock dividers, stopping modules, power gating control, selecting operating power control mode in Normal mode, and transitioning to low power modes.

[Table 10.1](#) lists the specifications of the low power mode functions. [Table 10.2](#) to [Table 10.4](#) lists the conditions to transition to low power modes, the states of the CPU and peripheral modules, and the method for canceling each mode. After a reset, the MCU enters the program execution state, but only the DMAC, DTC and SRAM operate.

**Table 10.1 Specifications of the low power mode functions**

Item	Specification
Reducing power consumption by switching clock signals	The frequency division ratio can be selected independently for the CPU clock (CPUCLK), system clock (ICLK), peripheral module clocks (PCLKA, PCLKB, PCLKC, PCLKD, PCLKE), and flash interface clock (FCLK). *1
Module stop	Functions can be stopped independently for each peripheral module
Power gating control	This function can be controlled the power state of the power domain. <ul style="list-style-type: none"> <li>Control the turning On/OFF for the power domain</li> <li>Control the retention of specific circuits during power gating</li> </ul>
Processor low power modes	<ul style="list-style-type: none"> <li>CPU Sleep mode</li> <li>CPU Deep Sleep mode</li> </ul>
Low-power modes	<ul style="list-style-type: none"> <li>Software Standby mode*2</li> <li>Deep Software Standby mode 1, 2, 3*2</li> </ul>
Operating power control modes	<ul style="list-style-type: none"> <li>Power consumption can be reduced in Normal and Processor low power mode by selecting an appropriate operating power control mode according to the operating frequency.</li> <li>Two operating power control modes are available: <ul style="list-style-type: none"> <li>High-speed mode</li> <li>Low-speed mode*2</li> </ul> </li> </ul>
TrustZone Filter	Security and Privilege attribution can be set

Note 1. For details, see [section 8, Clock Generation Circuit](#)

Note 2. This mode is not supported in external VDD mode.

**Table 10.2 Operating state of processor low power mode**

Item	CPU Sleep mode	CPU Deep Sleep mode
Transition condition	WFI instruction after set CPU0.SCR. SLEEPDEEP = 0.	WFI instruction after set CPU0.SCR. SLEEPDEEP = 1
Canceling method	All interrupts. Any reset available in the mode.	Interrupts shown in <a href="#">Table 10.4</a> Any reset available in the mode.
State after cancellation by an interrupt	Program execution state (interrupt processing)	Program execution state (interrupt processing)
State after cancellation by a reset	Reset state	Reset state
CPU	Stop (Retained)	Stop (Retained)
DMA Controller (DMAC)	Selectable	Selectable
Data Transfer Controller (DTC)	Selectable	Selectable
Watchdog Timer (WDT)	Selectable*1	Selectable*1
Independent Watchdog Timer (IWDT)	Selectable*1	Selectable*1
ARM Debug function	Stop*2	Stop*2
Trace function	Stop*3	Stop*3
Other peripheral modules	—	—

Note: Selectable means that operating or not operating can be selected by setting the control registers before entering Processor Low Power Mode.

Stop (Retained) means that the contents of the internal registers are retained but the operations are suspended.

Stop (Undefined) means that the contents of the internal registers are undefined and power to the internal circuit is cut off.

“ — ” means that the operation is not affected.

Note 1. In IWDT, operating or stopping is selected by setting the IWDT Stop Control bit (IWDTSTPCTL) in Option Function Select register 0 (OFS0) in IWDT auto start mode.



In WDT, operating or stopping is selected by setting the WDT Stop Control bit (WDTSTPCTL) in Option Function Select Register 0 (OFS0) in WDT auto start mode. Also, operating or stopping is selected by setting the WDT.WDTCSTPR.SLCSTP bit in WDT register start mode.

Note 2. This function is operating if debugger is connecting (DBGSTR.CDBGPWRUPREQ = 1 and SYOCDRCR.DBGEN = 1) before entering this mode.

Note 3. This function is operating if debugger is connecting (DBGSTR.CDBGPWRUPREQ = 1 and SYOCDRCR.DBGEN = 1) and TRCKCR.TRCKEN = 1 before entering this mode.

**Table 10.3 Operating state of each low power mode (1 of 2)**

Item	Software Standby Mode(SSTBY)	Deep Software Standby mode(DSTBY)		
	SSTBY	DSTBY1	DSTBY2	DSTBY3
Transition condition	WFI instruction after set LPSCR and CPU0.SCR.SLEEPDEEP=1	WFI instruction after set LPSCR and CPU0.SCR.SLEEPDEEP=1.		
Canceling method	Interrupts shown in <a href="#">Table 10.4</a> Any reset available in the mode	Interrupts shown in <a href="#">Table 10.4</a> Any reset available in the mode		
State after cancellation by an interrupt	Program execution state (interrupt processing)	Reset state		
State after cancellation by a reset	Reset state	Reset state		
Main clock oscillator	Selectable* <sup>9</sup>	Stop		
Sub-clock oscillator	Selectable	Selectable		
High-speed on-chip oscillator	Selectable* <sup>10</sup>	Stop		
Middle-speed on-chip oscillator	Stop* <sup>18</sup>	Stop		
Low-speed on-chip oscillator	Selectable* <sup>2</sup>	Selectable* <sup>2</sup>	Stop	
PLL1	Stop	Stop		
PLL2	Stop	Stop		
Oscillation stop detection function	Selectable* <sup>11</sup>	Stop		
Clock/buzzer output function	Selectable* <sup>3</sup>	Stop (Undefined)		
CPU	Stop (Retained)	Stop (Undefined)		
TCM (SRAM)	Stop (Retained)* <sup>14</sup>	Stop (Undefined)		
User SRAM	Stop (Retained)* <sup>12</sup>	Stop (Undefined)		
Standby SRAM	Stop (Retained)* <sup>13</sup>	Stop (Retained)* <sup>13</sup>	Stop (Undefined)	
Backup register	Stop (Retained)	Stop (Retained)		
Flash memory	Stop (Retained)	Stop (Retained)		
Memory Protection Unit (MPU)	Stop (Retained)	Stop (Undefined)		
DMA Controller (DMAC)	Stop (Retained)	Stop (Undefined)		
Data Transfer Controller (DTC)	Stop (Retained)	Stop (Undefined)		
Watchdog Timer (WDT)	Stop (Retained)	Stop (Undefined)		
Independent Watchdog Timer (IWDT)	Selectable* <sup>1</sup>	Selectable* <sup>1</sup>	Stop (Undefined)	
ARM Debug function	Stop* <sup>15</sup>	Stop* <sup>15</sup>		
Trace function	Stop* <sup>16</sup>	Stop* <sup>16</sup>		
Clock Frequency Accuracy Measurement Circuit (CAC)	Stop (Undefined)	Stop (Undefined)		
Ethernet MAC Controller (ETHERC)	Stop (Undefined)	Stop (Undefined)		
Ethernet DMA Controller (EDMAC)	Stop (Undefined)	Stop (Undefined)		
USB 2.0 Full-Speed Module (USBFS)	Stop (Retained) Detection of USB resumption is possible.	Stop (Retained) Detection of USB resumption is possible.	Stop (Undefined)	
Realtime clock (RTC)	Selectable	Selectable	Selectable* <sup>4</sup>	
CAN-FD	Stop (Undefined)	Stop (Undefined)		
CANFD ECC (CNECC)	Stop (Undefined)	Stop (Undefined)		
Serial Peripheral Interface (SPI0)	Stop (Retained)	Stop (Undefined)		
Serial Peripheral Interface (SPI1)	Stop (Undefined)	Stop (Undefined)		

**Table 10.3 Operating state of each low power mode (2 of 2)**

Item	Software Standby Mode(SSTBY)	Deep Software Standby mode(DSTBY)		
	SSTBY	DSTBY1	DSTBY2	DSTBY3
Octa Serial Peripheral Interface (OSPI)	Stop (Retained)	Stop (Undefined)		
Serial Sound Interface Enhanced (SSIE0)	Stop (Retained)	Stop (Undefined)		
Serial Sound Interface Enhanced (SSIE1)	Stop (Undefined)	Stop (Undefined)		
Cyclic Redundancy Check (CRC) Calculator	Stop (Undefined)	Stop (Undefined)		
Port Output Enable for GPT (POEG)	Stop (Undefined)	Stop (Undefined)		
General PWM Timer (GPT)	Stop (Undefined)	Stop (Undefined)		
Ultra low power Timer (ULPTn, n = 0, 1)	Selectable	Selectable	Stop (Undefined)	
Asynchronous General Purpose Timer (AGTn, n = 0, 1)	Selectable <sup>*5</sup>	Stop (Undefined)		
12-Bit A/D Converter (ADC12)	Stop (Undefined)	Stop (Undefined)		
12-Bit D/A Converter (DAC12)	Stop (Retained)	Stop (Undefined)		
Data Operation Circuit (DOC)	Stop (Undefined)	Stop (Undefined)		
Serial Communications Interface (SCI0)	Stop (Retained)	Stop (Undefined)		
Serial Communications Interface (SCIn, n = 1 to 4, 9)	Stop (Undefined)	Stop (Undefined)		
I2C Bus Interface (IIC0)	Selectable <sup>*6</sup>	Stop (Undefined)		
I2C Bus Interface (IIC1)	Stop (Undefined)	Stop (Undefined)		
Event Link Controller (ELC)	Stop (Undefined)	Stop (Undefined)		
Renesas Secure IP (RSIP-E51A)	Stop (Retained)	Stop (Undefined)		
Capture Engine Unit(CEU)	Stop (Undefined)	Stop (Undefined)		
Temperature Sensor (TSN)	Stop (Undefined)	Stop (Undefined)		
High-Speed Analog Comparator 0 (ACMPHS0)	Selectable	Stop (Undefined)		
High-Speed Analog Comparator 1 (ACMPHS1)	Selectable <sup>*19</sup>	Stop (Undefined)		
IRQn (n = 0 to 15) pin interrupt	Selectable	Stop (Undefined)		
NMI, IRQn-DS (n = 0 to 15) pin interrupt	Selectable	Selectable		
Programmable Voltage Detect (PVD)	Selectable	Selectable <sup>*17</sup>	Selectable <sup>*17</sup>	Stop (Undefined) <sup>*7</sup>
VBATT_R voltage drop detection	Selectable	Selectable	Selectable	Stop (Undefined)
Power-on reset circuit	Operating	Operating	Operating	Operating <sup>*8</sup>
I/O Ports	Retained	Retained		

Note: Selectable means that operating or not operating can be selected by setting the control registers before low power mode.

Stop (Retained) means that the contents of the internal registers are retained but the operations are suspended.

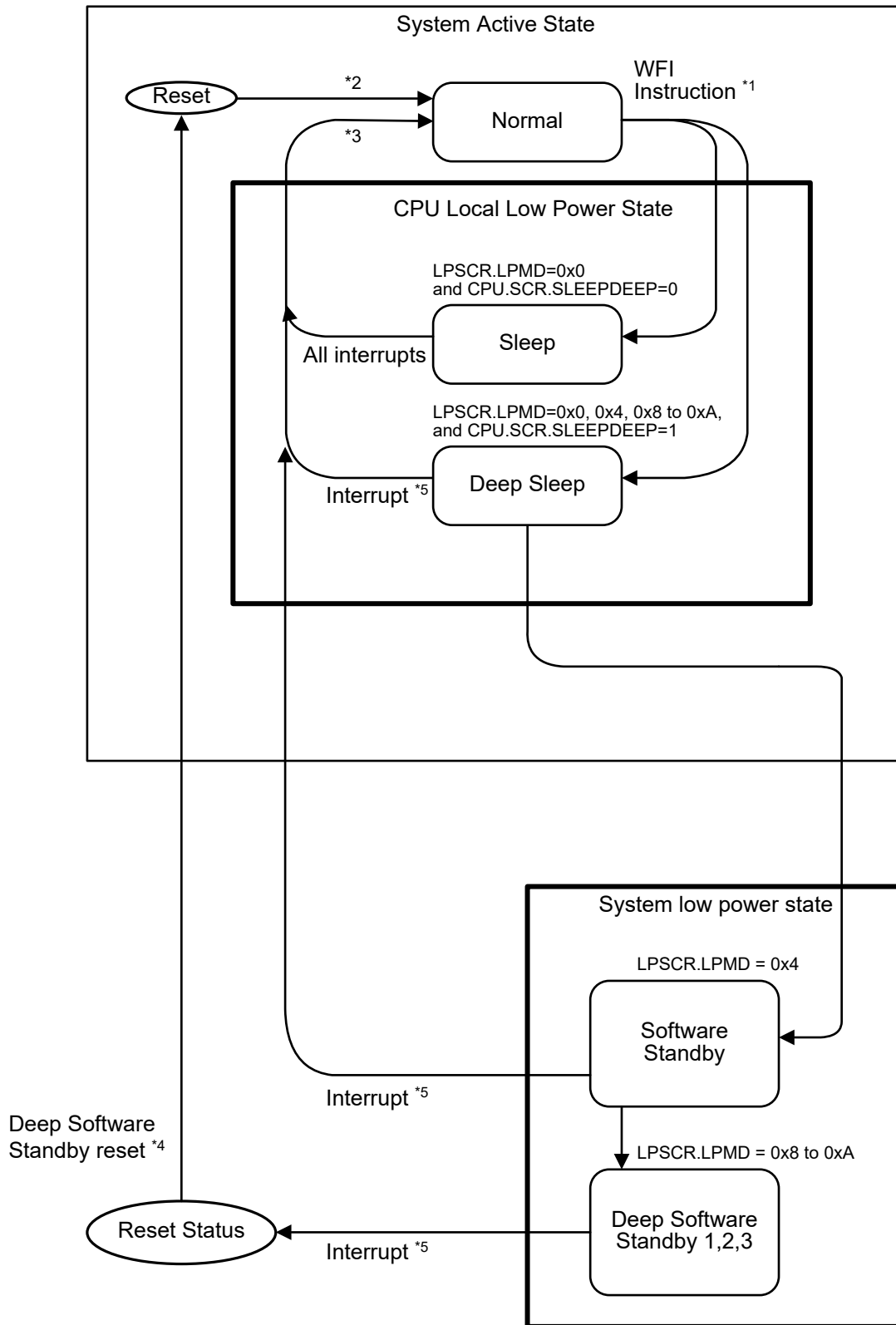
Stop (Undefined) means that the contents of the internal registers are undefined and power to the internal circuit is cut off. After canceling from each low power mode, the internal registers are initialized.

- Note 1. In IWDT, operating or stopping is selected by setting the IWDT Stop Control bit (IWDTSTPCTL) in Option Function Select register 0 (OFS0) in IWDT auto start mode. Also, operating or stopping is selected by setting the IWDT.IWDTCSTPR.SLCSTP bit in IWDT register start mode.
- Note 2. If IWDT is not used and LOCOCR.LCSTP = 0, LOCO is not stopped in Software Standby mode or Deep Software Standby mode 1. If IWDT is used and IWDT Stop Control bit is 0 (OFS.IWDTSTPCTL = 0 or IWDT.IWDTCSTPR.SLCSTP = 0), LOCO is not stopped regardless the value of LOCOCR.LCSTP in Software Standby mode or Deep Software Standby mode 1. If IWDT is used and IWDT Stop Control bit is 1 (OFS.IWDTSTPCTL = 1 or IWDT.IWDTCSTPR.SLCSTP = 1), LOCO is not stopped with LOCOCR.LCSTP = 0 in Software Standby mode or Deep Software Standby mode 1. In other cases, LOCO is stopped in Software Standby mode or Deep Software Standby mode 1.
- Note 3. Stopped when the clock output source select bits (CKOCR.CKOSEL[2:0]) are set to a value other than 010b (LOCO) and 100b (SOSC).
- Note 4. Only sub clock oscillator can be selected as the count source for RTC. When the RCR4.RCKSEL bit set to 1 (LOCO), the LPSCR must set to Deep Software Standby mode 1 before entering Deep Software Standby mode.
- Note 5. AGT0 operation is possible when 100b (AGTLCLK) or 110b (AGTSCLK) is selected by the AGT0.AGTMR1.TCK[2:0] bits. AGT1 operation is possible when 100b (AGTLCLK), 110b (AGTSCLK) or 101b (Underflow event signal from AGT0) is selected by the AGT1.AGTMR1.TCK[2:0] bits. keeping the output state or changing to the
- Note 6. Only IIC0 wakeup function is available.

- Note 7. When using PVD in Deep Software Standby mode, LPSCR must be set to DSTBY1 or DSTBY2 before entering Deep Software Standby mode.
- Note 8. When the MCU enters Deep Software Standby mode 3, the PVD circuit stops and the low-power function of the power-on reset circuit is enabled.
- Note 9. When MOSCSCR.MOSCSOKP = 1 while MOSC is oscillated, MOSC continues to oscillate in Software Standby mode.
- Note 10. When HOCOSCR.HOCOSOKP = 1 while HOCO is oscillated, HOCO continues to oscillate in Software Standby mode.
- Note 11. The function is depending on the setting value of the MOSCSCR.MOSCSOKP bit.
- Note 12. If PDRAMSCR0.RKEEPn bit is set to 0, the contents of the target User SRAM is not retained.
- Note 13. If DPSBYCR.SRKEEP bit is set to 0, the contents of Standby RAM is not retained.
- Note 14. If PDRAMSCR1.RKEEPn bit is set to 0, the contents of the target TCM is not retained.
- Note 15. This function is operating if debugger is connecting before entering this mode.
- Note 16. This function is operating if debugger is connecting and TRCKCR.TRCKEN = 1 before entering this mode.
- Note 17. When OFS1(\_SEC).PVDAS = 0 and OFS1(\_SEC).PVDLPSEL = 0, low power consumption function of PVD0 is enabled during DSTBY1 and DSTBY2. When OFS1(\_SEC).PVDAS = 0 and OFS1(\_SEC).PVDLPSEL = 1, low power consumption function of PVD0 is disabled during DSTBY1 and DSTBY2. See [section 52, Electrical Characteristics](#).
- Note 18. When the on-chip debugger function is valid, MOCO does not stop in this mode.
- Note 19. Only VCOUT function is permitted. The VCOUT pin operates when ACMPHS uses no digital filter. For details on digital filter, see [section 45, High-Speed Analog Comparator \(ACMPHS\)](#).

**Table 10.4 Interrupt source for canceling CPU Deep Sleep, Software Standby and Deep Software Standby modes**

Interrupt source	Name	CPU Deep Sleep Mode	Software Standby Mode	Deep Software Standby mode		
				DSTBY1	DSTBY2	DSTBY3
NMI		Yes	Yes	Yes	Yes	Yes
Port	PORT_IRQn (n = 0 to 15)	Yes	Yes	No	No	No
	PORT_IRQn-DS (n = 0 to 15)	Yes	Yes	Yes	Yes	Yes
PVD	PVD_PVDm (m = 1, 2)	Yes	Yes	Yes	Yes	No
IWDT	IWDT_NMIUNDF	Yes	Yes	Yes	No	No
USBFS	USBFS_USBR	Yes	Yes	Yes	No	No
RTC	RTC_ALM	Yes	Yes	Yes	Yes	Yes
	RTC_PRD	Yes	Yes	Yes	Yes	Yes
ULPT0	ULPT0_ULPTI	Yes	Yes	Yes	No	No
	ULPT0_ULPTCMAI	Yes	Yes	No	No	No
	ULPT0_ULPTCMBI	Yes	Yes	No	No	No
ULPT1	ULPT1_ULPTI	Yes	Yes	Yes	No	No
	ULPT1_ULPTCMAI	Yes	Yes	No	No	No
	ULPT1_ULPTCMBI	Yes	Yes	No	No	No
AGT1	AGT1_AGTI	Yes	Yes	No	No	No
	AGT1_AGTCMAI	Yes	Yes	No	No	No
	AGT1_AGTCMBI	Yes	Yes	No	No	No
ACMPHS0	ACMP_HS0	Yes	Yes	No	No	No
IIC0	IIC0_WUI	Yes	Yes	No	No	No
BBF (Battery Backup Function)	VBATT_TADI	Yes	Yes	Yes	Yes	Yes



Note 1. When an interrupt as a trigger for cancel is received during a transition to the program-stopped state after the execution of a WFI instruction, the MCU executes interrupt exception handling instead of transitioning to low power mode. When entering to Deep Software Standby mode, if an interrupt occurs after the state has transitioned, after mode transition is completed, returning with Deep Software Standby reset.

- Note 2. The MOCO clock is the source of the operating clock following a transition from the reset state to Normal mode.
- Note 3. When the transition to Normal mode is done because of an interrupt from CPU Sleep mode, CPU Deep Sleep mode or Software Standby mode, the clock source is the same as before entering the each low power modes.
- Note 4. When an available interrupt request is generated, an internal reset (Deep Software Standby reset) is generated over a fixed period. Canceling of Deep Software Standby mode accompanies release from the internal reset state, and then the MCU transitions to Normal mode and executes reset exception processing with the MOCO clock as the source of the operating clock.
- Note 5. For interrupt sources for canceling each low power modes, see [Table 10.4](#)

**Figure 10.1 Mode Transitions**

## 10.2 Register Descriptions

### 10.2.1 LPMSAR : Low Power Mode Security Attribution Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x3C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	NONS EC21	—	NONS EC19	NONS EC18	NONS EC17	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	NONS EC8	—	—	—	—	—	NONS EC2	NONS EC1	NONS EC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non-secure Attribute bit 0 Target register: OPCCR 0: Secure 1: Non-secure	R/W
1	NONSEC1	Non-secure Attribute bit 1 Target register: LPSCR, DPSWCR 0: Secure 1: Non-secure	R/W
2	NONSEC2	Non-secure Attribute bit 2 Target register: SBYCR, SSCR1 0: Secure 1: Non-secure	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	NONSEC8	Non-secure Attribute bit 8 Target register: DPSBYCR 0: Secure 1: Non-secure	R/W
16:9	—	These bits are read as 0. The write value should be 0.	R/W
17	NONSEC17	Non-secure Attribute bit 17 Target register: HOCOLDOCR 0: Secure 1: Non-secure	R/W
18	NONSEC18	Non-secure Attribute bit 18 Target register: PLL1LDOCR 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
19	NONSEC19	Non-secure Attribute bit 19 Target register: PLL2LDOCR 0: Secure 1: Non-secure	R/W
20	—	This bit is read as 0. The write value should be 0.	R/W
21	NONSEC21	Non-secure Attribute bit 21 Target register: LVOCR 0: Secure 1: Non-secure	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

Note: Set the PRCR.PRC4 bit to 1 (write enabled) before rewriting this register.

The LPMSAR register controls the secure attribute of Low Power Mode registers.

#### NONSEC0 bit (Non-secure Attribute bit 0)

This bit controls the security attribute of OPCCR.

#### NONSEC1 bit (Non-secure Attribute bit 1)

This bit controls the security attribute of LPSCR, DPSECR.

#### NONSEC2 bit (Non-secure Attribute bit 2)

This bit controls the security attribute of SBYCR, SSCR1.

#### NONSEC8 bit (Non-secure Attribute bit 8)

This bit controls the security attribute of DPSBYCR.

#### NONSEC17 bit (Non-secure Attribute bit 17)

This bit controls the security attribute of HOCOLDOCR.

#### NONSEC18 bit (Non-secure Attribute bit 18)

This bit controls the security attribute of PLL1LDOCR.

#### NONSEC19 bit (Non-secure Attribute bit 19)

This bit controls the security attribute of PLL2LDOCR.

#### NONSEC21 bit (Non-secure Attribute bit 21)

This bit controls the security attribute of LVOCR.

### 10.2.2 DPFSAR : Deep Software Standby Interrupt Factor Security Attribution Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x3E0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DPFS A31	—	DPFS A29	—	DPFS A27	DPFS A26	—	DPFS A24	—	—	—	DPFS A20	DPFS A19	DPFS A18	DPFS A17	DPFS A16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DPFS A15	DPFS A14	DPFS A13	DPFS A12	DPFS A11	DPFS A10	DPFS A9	DPFS A8	DPFS A7	DPFS A6	DPFS A5	DPFS A4	DPFS A3	DPFS A2	DPFS A1	DPFS A0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DPFSA7 to DPFSA0	Deep Software Standby Interrupt Factor Security Attribute bit n (n = 0 to 7) Target register: DPSIER0.bn, DPSIFR0.bn, DPSIEGR0.bn (n = 0 to 7) Target factor : IRQn-DS Pin (n = 0 to 7) 0: Secure 1: Non-secure	R/W
15:8	DPFSA15 to DPFSA8	Deep Software Standby Interrupt Factor Security Attribute bit n (n = 8 to 15) Target register: DPSIER1.bn, DPSIFR1.bn, DPSIEGR1.bn (n = 0 to 7) Target factor : IRQn-DS Pin (n = 8 to 15) 0: Secure 1: Non-secure	R/W
16	DPFSA16	Deep Software Standby Interrupt Factor Security Attribute bit 16 Target register: DPSIER2.b0, DPSIFR2.b0, DPSIEGR2.b0 Target factor : PVD1 0: Secure 1: Non-secure	R/W
17	DPFSA17	Deep Software Standby Interrupt Factor Security Attribute bit 17 Target register: DPSIER2.b1, DPSIFR2.b1, DPSIEGR2.b1 Target factor : PVD2 0: Secure 1: Non-secure	R/W
18	DPFSA18	Deep Software Standby Interrupt Factor Security Attribute bit 18 Target register: DPSIER2.b2, DPSIFR2.b2 Target factor : RTC Interval 0: Secure 1: Non-secure	R/W
19	DPFSA19	Deep Software Standby Interrupt Factor Security Attribute bit 19 Target register: DPSIER2.b3, DPSIFR2.b3 Target factor : RTC Alarm 0: Secure 1: Non-secure	R/W
20	DPFSA20	Deep Software Standby Interrupt Factor Security Attribute bit 20 Target register: DPSIER2.b4, DPSIFR2.b4, DPSIEGR2.b4 Target factor : NMI Pin 0: Secure 1: Non-secure	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	DPFSA24	Deep Software Standby Interrupt Factor Security Attribute bit 24 Target register: DPSIER3.b0, DPSIFR3.b0 Target factor : USBFS Suspend/Resume 0: Secure 1: Non-secure	R/W
25	—	This bit is read as 0. The write value should be 0.	R/W
26	DPFSA26	Deep Software Standby Interrupt Factor Security Attribute bit 26 Target register: DPSIER3.b2, DPSIFR3.b2 Target factor : ULPT0 0: Secure 1: Non-secure	R/W
27	DPFSA27	Deep Software Standby Interrupt Factor Security Attribute bit 27 Target register: DPSIER3.b3, DPSIFR3.b3 Target factor : ULPT1 0: Secure 1: Non-secure	R/W
28	—	This bit is read as 0. The write value should be 0.	R/W
29	DPFSA29	Deep Software Standby Interrupt Factor Security Attribute bit 29 Target register: DPSIER3.b5, DPSIFR3.b5 Target factor : IWDT Underflow 0: Secure 1: Non-secure	R/W
30	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31	DPFSA31	Deep Software Standby Interrupt Factor Security Attribute bit 31 Target register: DPSIER3.b7, DPSIFR3.b7 Target factor : Tamper Detection 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1

Note: Set the PRCR.PRC4 bit to 1 (write enabled) before rewriting this register.

The DPFSA register controls the secure attribute of Deep Software Standby Interrupt Factor control registers.

#### **DPFSA bit (Deep Software Standby Interrupt Factor Security Attribute bit n (n = 0 to 7))**

This bit controls the security attribute of DPSIER0.bn, DPSIFR0.bn, DPSIEGR0.bn (n = 0 to 7) .

Target factor is IRQn-DS Pin (n = 0 to 7).

#### **DPFSA bit (Deep Software Standby Interrupt Factor Security Attribute bit n (n = 8 to 15))**

This bit controls the security attribute of DPSIER1.bn, DPSIFR1.bn, DPSIEGR1.bn (n = 0 to 7) .

Target factor is IRQn-DS Pin (n = 8 to 15).

#### **DPFSA16 bit (Deep Software Standby Interrupt Factor Security Attribute bit 16)**

This bit controls the security attribute of DPSIER2.b0, DPSIFR2.b0, DPSIEGR2.b0 .

Target factor is PVD1.

#### **DPFSA17 bit (Deep Software Standby Interrupt Factor Security Attribute bit 17)**

This bit controls the security attribute of DPSIER2.b1, DPSIFR2.b1, DPSIEGR2.b1.

Target factor is PVD2.

#### **DPFSA18 bit (Deep Software Standby Interrupt Factor Security Attribute bit 18)**

This bit controls the security attribute of DPSIER2.b2, DPSIFR2.b2.

Target factor is RTC Interval.

#### **DPFSA19 bit (Deep Software Standby Interrupt Factor Security Attribute bit 19)**

This bit controls the security attribute of DPSIER2.b3, DPSIFR2.b3.

Target factor is RTC Alarm.

#### **DPFSA20 bit (Deep Software Standby Interrupt Factor Security Attribute bit 20)**

This bit controls the security attribute of DPSIER2.b4, DPSIFR2.b4, DPSIEGR2.b4.

Target factor is NMI Pin.

#### **DPFSA24 bit (Deep Software Standby Interrupt Factor Security Attribute bit 24)**

This bit controls the security attribute of DPSIER3.b0, DPSIFR3.b0.

Target factor is USBFS Suspend/Resume.

#### **DPFSA26 bit (Deep Software Standby Interrupt Factor Security Attribute bit 26)**

This bit controls the security attribute of DPSIER3.b2, DPSIFR3.b2.

Target factor is ULPT0.

#### **DPFSA27 bit (Deep Software Standby Interrupt Factor Security Attribute bit 27)**

This bit controls the security attribute of DPSIER3.b3, DPSIFR3.b3.

Target factor is ULPT1.

#### **DPFSA29 bit (Deep Software Standby Interrupt Factor Security Attribute bit 29)**

This bit controls the security attribute of DPSIER3.b5, DPSIFR3.b5.



Target factor is IWDT Underflow.

**DPFSA31 bit (Deep Software Standby Interrupt Factor Security Attribute bit 31)**

This bit controls the security attribute of DPSIER3.b7, DPSIFR3.b7.

Target factor is Tamper Detection.

**10.2.3 RSCSAR : RAM Standby Control Security Attribution Register**

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0x3E4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSCS A16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RSCS A6	RSCS A5	RSCS A4	RSCS A3	RSCS A2	RSCS A1	RSCS A0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RSCSA2 to RSCSA0	Unnecessary Circuit Control Security Attribution bit n (n = 0 to 2) Target register: PDRAMSCR0.bn (n = 0 to 2) 0: Secure 1: Non-secure	R/W
6:3	RSCSA6 to RSCSA3	RAM Standby Control Security Attribute bit n (n = 3 to 6) Target register: PDRAMSCR0.bn (n = 3 to 6) 0: Secure 1: Non-secure	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W
16	RSCSA16	RAM Standby Control Security Attribute bit n (n = 16) Target register: PDRAMSCR1.bn (n = 0) 0: Secure 1: Non-secure	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

Note: Set the PRCR.PRC4 bit to 1 (write enabled) before rewriting this register.

The RSCSAR register controls the secure attribute of RAM Standby Control registers.

**RSCSAn (n = 0 to 2) bit (Unnecessary Circuit Control Security Attribution bit n)**

This bit controls the security attribute of PDRAMSCR0.bn (n = 0 to 2).

**RSCSAn (n = 3 to 6) bit (RAM Standby Control Security Attribute bit n)**

This bit controls the security attribute of PDRAMSCR0.bn (n = 3 to 6).

**RSCSAn (n = 16) bit (RAM Standby Control Security Attribute bit n)**

This bit controls the security attribute of PDRAMSCR1.bn (n = 0).

### 10.2.4 MSTPCRA : Module Stop Control Register A

Base address: MSTP = 0x4020\_3000  
 MSTP\_NS = 0x5020\_3000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	MSTP A22	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSTP A15	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTP A1	MSTP A0
Value after reset:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0

Bit	Symbol	Function	R/W
0	MSTPA0	Unnecessary Circuit Module Stop <sup>*2*3</sup> Target module: Unnecessary Circuit. 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
1	MSTPA1	SRAM1 Module Stop <sup>*2</sup> Target module: SRAM1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14:2	—	These bits are read as 1. The write value should be 1.	R/W
15	MSTPA15	Standby SRAM Module Stop Target module: Standby SRAM 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21:16	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPA22	DMA Controller and Data Transfer Controller Module Stop <sup>*1</sup> Target module: DMAC, DTC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:23	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-4, P-TYPE-2

Note 1. When rewriting the MSTPA22 bit from 0 to 1, disable the DMAC and DTC before setting the MSTPA22 bit.

Note 2. When changing the value of this bit, only execute subsequent instructions after reading this bit to check that the value was updated.

Note 3. This bit has an initial value of 0. Set this bit to 1 to reduce power consumption.

### 10.2.5 MSTPCRB : Module Stop Control Register B

Base address: MSTP = 0x4020\_3000  
 MSTP\_NS = 0x5020\_3000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTP B31	MSTP B30	MSTP B29	MSTP B28	MSTP B27	—	—	—	—	MSTP B22	—	—	MSTP B19	MSTP B18	—	MSTP B16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSTP B15	—	—	—	MSTP B11	—	MSTP B9	MSTP B8	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 1. The write value should be 1.	R/W
8	MSTPB8	I <sup>2</sup> C Bus Interface 1 Module Stop Target module: IIC1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
9	MSTPB9	I <sup>2</sup> C Bus Interface 0 Module Stop Target module: IIC0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
10	—	This bit is read as 1. The write value should be 1.	R/W
11	MSTPB11	Universal Serial Bus 2.0 FS Interface Module Stop* <sup>1</sup> Target module: USBFS 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14:12	—	These bits are read as 1. The write value should be 1.	R/W
15	MSTPB15	ETHERC0 and EDMAC0 Controller Module Stop Target module: ETHERC0 and EDMAC0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
16	MSTPB16	Octal Serial Peripheral Interface Module Stop* <sup>2</sup> Target module: OSP1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
17	—	This bit is read as 1. The write value should be 1.	R/W
18	MSTPB18	Serial Peripheral Interface 1 Module Stop* <sup>3</sup> Target module: SPI1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
19	MSTPB19	Serial Peripheral Interface 0 Module Stop* <sup>3</sup> Target module: SPI0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21:20	—	These bits are read as 1. The write value should be 1.	R/W
22	MSTPB22	Serial Communication Interface 9 Module Stop* <sup>4</sup> Target module: SC19 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
26:23	—	These bits are read as 1. The write value should be 1.	R/W
27	MSTPB27	Serial Communication Interface 4 Module Stop* <sup>4</sup> Target module: SC14 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPB28	Serial Communication Interface 3 Module Stop* <sup>4</sup> Target module: SC13 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
29	MSTPB29	Serial Communication Interface 2 Module Stop* <sup>4</sup> Target module: SC12 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
30	MSTPB30	Serial Communication Interface 1 Module Stop* <sup>4</sup> Target module: SC11 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
31	MSTPB31	Serial Communication Interface 0 Module Stop*4 Target module: SCIO 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note: S-TYPE-4, P-TYPE-4

- Note 1. The MSTPBi bit must be written while the oscillation of the clock controlled by this bit is stabilized. For entering Software Standby mode after writing the MSTPBi bit, wait for two USB clock (USBCLK) cycles after writing, and then execute a WFI instruction (i = 11 to 12).
- Note 2. The MSTPB16 bit must be written while the oscillation of the clock controlled by this bit is stabilized. For entering Software Standby mode after writing the MSTPB16 bit, wait for two Octal-SPI clock (OCTACLK) cycles after writing, and then execute a WFI instruction. When changing the value of this bit, only execute subsequent instructions after reading this bit to check that the value was updated.
- Note 3. The MSTPBi bit must be written while the oscillation of the clock controlled by this bit is stabilized. For entering Software Standby mode after writing the MSTPBi bit, wait for two SPI clock (SPICLK) cycles after writing, and then execute a WFI instruction (i = 18 to 19).
- Note 4. The MSTPBi bit must be written while the oscillation of the clock controlled by this bit is stabilized. For entering Software Standby mode after writing the MSTPBi bit, wait for two SCI clock (SCICLK) cycles after writing, and then execute a WFI instruction (i = 22, 27 to 31).

### 10.2.6 MSTPCRC : Module Stop Control Register C

Base address: MSTP = 0x4020\_3000  
MSTP\_NS = 0x5020\_3000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTP C31	—	—	—	MSTP C27	MSTP C26	—	—	—	—	—	—	—	—	—	MSTP C16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MSTP C14	MSTP C13	—	—	—	—	MSTP C8	MSTP C7	—	—	—	—	—	MSTP C1	MSTP C0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPC0	Clock Frequency Accuracy Measurement Circuit Module Stop*1 Target module: CAC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
1	MSTPC1	Cyclic Redundancy Check Calculator Module Stop Target module: CRC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
6:2	—	These bits are read as 1. The write value should be 1.	R/W
7	MSTPC7	Serial Sound Interface Enhanced 1 Module Stop Target module: SSIE1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
8	MSTPC8	Serial Sound Interface Enhanced 0 Module Stop Target module: SSIE0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12:9	—	These bits are read as 1. The write value should be 1.	R/W
13	MSTPC13	Data Operation Circuit Module Stop Target module: DOC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
14	MSTPC14	Event Link Controller Module Stop Target module: ELC 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15	—	This bit is read as 1. The write value should be 1.	R/W
16	MSTPC16	Capture Engine Unit Module Stop Target module: CEU 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
25:17	—	These bits are read as 1. The write value should be 1.	R/W
26	MSTPC26	Controller Area Network with Flexible Data-Rate 1 Module Stop* <sup>2</sup> Target module: CANFD1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
27	MSTPC27	Controller Area Network with Flexible Data-Rate 0 Module Stop* <sup>2</sup> Target module: CANFD0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
30:28	—	These bits are read as 1. The write value should be 1.	R/W
31	MSTPC31	Renesas Secure IP Module Stop Target module: RSIP-E51A 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note: S-TYPE-4, P-TYPE-4

Note 1. The MSTPC0 bit must be written while the oscillation of the clock to be controlled by this bit is stable. To enter Software Standby mode after writing this bit, wait for 2 cycles of the slowest clock from the clocks output by the oscillators, then execute a WFI instruction.

Note 2. The MSTPC<sub>i</sub> bit must be written while the oscillation of the clock controlled by this bit is stabilized. For entering Software Standby mode after writing the MSTPC<sub>i</sub> bit, wait for two CANFD clock (CANFDCLK) cycles after writing, and then execute a WFI instruction( = 26 to 27)

### 10.2.7 MSTPCRD : Module Stop Control Register D

Base address: MSTP = 0x4020\_3000  
MSTP\_NS = 0x5020\_3000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	MSTP D28	MSTP D27	—	—	—	—	MSTP D22	—	MSTP D20	—	—	—	MSTP D16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSTP D15	MSTP D14	MSTP D13	MSTP D12	MSTP D11	—	—	—	—	—	MSTP D5	MSTP D4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 1. The write value should be 1.	R/W
4	MSTPD4	Low Power Asynchronous General Purpose Timer 1 Module Stop* <sup>1</sup> Target module: AGT1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
5	MSTPD5	Low Power Asynchronous General Purpose Timer 0 Module Stop* <sup>1</sup> Target module: AGT0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
10:6	—	These bits are read as 1. The write value should be 1.	R/W
11	MSTPD11	Port Output Enable for GPT Group D Module Stop Target module: POEG Group D 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
12	MSTPD12	Port Output Enable for GPT Group C Module Stop Target module: POEG Group C 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
13	MSTPD13	Port Output Enable for GPT Group B Module Stop Target module: POEG Group B 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
14	MSTPD14	Port Output Enable for GPT Group A Module Stop Target module: POEG Group A 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
15	MSTPD15	12-bit A/D Converter 1 Module Stop Target module: ADC12_1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
16	MSTPD16	12-bit A/D Converter 0 Module Stop Target module: ADC12_0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
19:17	—	These bits are read as 1. The write value should be 1.	R/W
20	MSTPD20	12-bit D/A Converter Module Stop Target module: DAC12 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21	—	This bit is read as 1. The write value should be 1.	R/W
22	MSTPD22	Temperature Sensor Module Stop Target module: TSN 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
26:23	—	These bits are read as 1. The write value should be 1.	R/W
27	MSTPD27	High-Speed Analog Comparator 1 Module Stop Target module: ACMPHS1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPD28	High-Speed Analog Comparator 0 Module Stop Target module: ACMPHS0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31:29	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-4, P-TYPE-4

Note 1. When the count source is sub-clock oscillator or LOCO, AGT<sub>n</sub> counting does not stop even if MSTPD<sub>i</sub> is set to 1. If the count source is the sub-clock oscillator or LOCO, this bit must be set to 1 except when accessing the AGT<sub>n</sub> registers. (n = 0, 1) (i = 4, 5)

## 10.2.8 MSTPCRE : Module Stop Control Register E

Base address: MSTP = 0x4020\_3000  
MSTP\_NS = 0x5020\_3000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSTP E31	MSTP E30	MSTP E29	MSTP E28	MSTP E27	MSTP E26	—	—	—	—	MSTP E21	MSTP E20	MSTP E19	MSTP E18	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MSTP E9	MSTP E8	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 1. The write value should be 1.	R/W
8	MSTPE8	Ultra-Low Power Timer 1 Module Stop* <sup>1</sup> Target module: ULPT1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
9	MSTPE9	Ultra-Low Power Timer 0 Module Stop* <sup>1</sup> Target module: ULPT0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
17:10	—	These bits are read as 1. The write value should be 1.	R/W
18	MSTPE18	General PWM Timer 13 Module* <sup>2</sup> Target module: GPT13 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
19	MSTPE19	General PWM Timer 12 Module* <sup>2</sup> Target module: GPT12 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
20	MSTPE20	General PWM Timer 11 Module* <sup>2</sup> Target module: GPT11 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
21	MSTPE21	General PWM Timer 10 Module* <sup>2</sup> Target module: GPT10 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
25:22	—	These bits are read as 1. The write value should be 1.	R/W
26	MSTPE26	General PWM Timer 5 Module* <sup>2</sup> Target module: GPT5 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
27	MSTPE27	General PWM Timer 4 Module* <sup>2</sup> Target module: GPT4 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
28	MSTPE28	General PWM Timer 3 Module* <sup>2</sup> Target module: GPT3 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Bit	Symbol	Function	R/W
29	MSTPE29	General PWM Timer 2 Module* <sup>2</sup> Target module: GPT2 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
30	MSTPE30	General PWM Timer 1 Module* <sup>2</sup> Target module: GPT1 0: Cancel the module-stop state 1: Enter the module-stop state	R/W
31	MSTPE31	General PWM Timer 0 Module* <sup>2</sup> Target module: GPT0 0: Cancel the module-stop state 1: Enter the module-stop state	R/W

Note: S-TYPE-4, P-TYPE-4

Note 1. When the count source is sub-clock oscillator or LOCO, ULPTn counting does not stop even if MSTPEi is set to 1. If the count source is the sub-clock oscillator or LOCO, this bit must be set to 1 except when accessing the ULPTn registers. (n = 1, 0) (i = 8, 9)

Note 2. The MSTPEi bit must be written while the oscillation of the clock controlled by this bit is stabilized. For entering Software Standby mode after writing the MSTPEi bit, wait for two GPT clock (GPTCLK) cycles after writing, and then execute a WFI instruction (i = 18 to 21, 26 to 31).

### 10.2.9 OPCCR : Operating Power Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x0A0

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	—	OPCM TSF	—	—	OPCM[1:0]
------------	---	---	---	-------------	---	---	-----------

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
1:0	OPCM[1:0]	Operating Power Control Mode Select 0 0: High-speed mode 0 1: Setting prohibited 1 0: Setting prohibited 1 1: Low-speed mode	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	OPCMTSF	Operating Power Control Mode Transition Status Flag 0: Transition completed 1: During transition	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The OPCCR register is used to reduce power consumption in Normal and Processor low power mode by specifying a lower operating frequency. For the procedure to change the operating power control modes, see [section 10.5. Function for Lower Operating Power Consumption](#).

When transitioning from Software Standby mode to Normal mode, the setting in the OPCCR.OPCM[1:0] bits are as follows, regardless of the setting before entering Software Standby mode:

- OPCCR.OPCM[1:0] = 00b (High-speed mode)

If Software Standby mode is canceled by interrupt before the transition to Software Standby completes, the OPCCR.OPCM[1:0] bits retain the setting from before the WFI instruction is executed. If this causes any problem, set the MCU to High-speed mode during the exception handling procedure when canceling Software Standby mode.



**OPCM[1:0] bits (Operating Power Control Mode Select)**

The OPCM[1:0] bits select the operating power control mode in Normal and Processor low power mode. Table 10.5 shows the relationship between the operating power control modes and the OPCM[1:0] settings.

**OPCMTSF flag (Operating Power Control Mode Transition Status Flag)**

The OPCMTSF flag indicates the switching control state when the operating power control mode is switched. This flag becomes 1 when the OPCM bit is written, and 0 when mode transition completes. Read this flag and confirm that it is 0 before proceeding.

**Table 10.5 The operating power control modes**

Operating power control mode	OPCM[1:0] bits	Power consumption
High-speed mode	00b	High
Low-speed mode	11b	Low

Operating frequency range and voltage range are shown in Electrical characteristics.

Each operating power control mode is described below.

- High-speed mode  
After a reset cancellation, the MCU is activated in this mode.
- Low-speed mode  
The following restrictions apply in Low-speed mode:
  - P/E operations for flash memory are prohibited.
  - Using the PLL1/PLL2 is prohibited. See section 10.7.1. Register Access
  - Writing to PLL1LDOCR, PLL2LDOCR and HOCOLDOCR is prohibited.

In this mode, lower power consumption is possible than in high-speed mode when the same operation is performed under the same conditions (operating frequency, operating voltage).

**10.2.10 PDRAMSCR0 : SRAM Power Domain Standby Control Register 0**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x140

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RKEE P6	RKEE P5	RKEE P4	RKEE P3	RKEE P2	RKEE P1	RKEE P0
Value after reset:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
2:0	RKEEP2 to RKEEP0	Unnecessary Circuit Retention*1 0: When entering the Software Standby mode, the contents of unnecessary circuit are not kept. 1: When entering the Software Standby mode, the contents of unnecessary circuit are kept.	R/W
6:3	RKEEP6 to RKEEP3	RAM Retention 0: When entering the Software Standby mode, the contents of the target RAM are not kept. 1: When entering the Software Standby mode, the contents of the target RAM are kept.	R/W
14:7	—	These bits are read as 1. The write value should be 1.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. This bit has an initial value of 0. Set this bit to 1 to reduce power consumption.

The PDRAMSCR0 register controls the target memory in Software Standby mode.

**RKEEPn bit (RAM Retention) (n = 0 to 6)**

The RKEEPn (n = 0 to 6) 7bits controls whether to retain the contents of the target memory or unnecessary circuit in Software Standby mode.

Table 10.6 shows the relationship between each bit and the target memory.

**Table 10.6 Relationship between each bit and the target memory**

PDRAMSCR0 register bit	Target memory
PDRAM.b3	SRAM1: 0x2206_0000 to 0x2207_FFFF (Secure alias) 0x3206_0000 to 0x3207_FFFF (Non-secure alias)
PDRAM.b4	SRAM1: 0x2208_0000 to 0x2209_FFFF (Secure alias) 0x3208_0000 to 0x3209_FFFF (Non-secure alias)
PDRAM.b5	SRAM1: 0x220A_0000 to 0x220B_FFFF (Secure alias) 0x320A_0000 to 0x320B_FFFF (Non-secure alias)
PDRAM.b6	SRAM1: 0x220C_0000 to 0x220D_FFFF (Secure alias) 0x320C_0000 to 0x320D_FFFF (Non-secure alias)

**10.2.11 PDRAMSCR1 : SRAM Power Domain Standby Control Register 1**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x142

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RKEEP0

Value after reset: 0 0 0 0 0 0 0 1 1

Bit	Symbol	Function	R/W
0	RKEEP0	RAM Retention 0: When entering the CPU Deep Sleep and Software Standby mode, the contents of the target RAM are not kept. 1: When entering the CPU Deep Sleep and Software Standby mode, the contents of the target RAM are kept.	R/W
1	—	This bit is read as 1. The write value should be 1.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The PDRAMSCR1 register controls the target memory in CPU Deep Sleep and Software Standby mode.

**RKEEP0 bit (RAM Retention)**

The RKEEP0 bit controls whether to retain the contents of the target memory in CPU Deep Sleep and Software Standby mode.

Table 10.7 shows the relationship between each bit and the target memory.

**Table 10.7 Relationship between each bit and the target memory**

PDRAM register bit	Target memory
PDRAM.b0	ITCM and DTCM

### 10.2.12 SSCR1 : Software Standby Control Register 1

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA98

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SS1FR

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	SS1FR	Software Standby Fast Return 0: When returning from Software Standby mode, fast return function is disabled 1: When returning from Software Standby mode, fast return function is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

#### SS1FR bit (Software Standby Fast Return)

When the SS1FR bit is 1, the recovery time from Software Standby mode is shorter.

Refer to the chapter of electrical characteristics on the detail of the recovery time.

Note: It is recommended to set this bit to 1. Otherwise it just increases the recovery time.

### 10.2.13 LPSCR : Low Power State Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA90

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	LPMD[3:0]			

Value after reset: 0 0 0 0 0 1 0 0 0

Bit	Symbol	Function	R/W
3:0	LPMD[3:0]	Low power mode setting bit 0x0: System Active 0x4: Software Standby mode 0x8: Deep Software Standby mode 1 0x9: Deep Software Standby mode 2 0xA: Deep Software Standby mode 3 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The LPSCR register controls the transition destination of low power mode.

LPSCR is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see [section 5, Resets](#).

#### LPMD[3:0] bit (Low power mode setting bit)

The LPMD[3:0] bit indicate the destination of low power mode. Actual low power mode transition is executed by WFI instruction in CPU.

The transition of Software Standby mode and Deep Software Standby mode does not occur unless the WFI instruction of each CPU is executed.

When the low power mode is canceled by an interrupt, The LPMD[3:0] bit remains the value before transition. Writing 0 to this bit clears the bit itself.

While FENTRYR.FENTRY<sub>i</sub> ( $i = 0$  to 3) is 1 or FENTRYR.FENTRYD bit is 1, the setting to Software Standby and Deep Software Standby mode is ineffective. Even if LPMD is any of 0x4, 0x8 to 0xA, the MCU is keep to System active state on execution of a WFI instruction. Refer to [Table 10.9](#) for the detail.

While OFS0.IWDTSTPCTL bit is 0 (counting continues), the setting to Deep software Standby 2 and 3 mode is ineffective. Even if LPMD is any of 0x9 or 0xA, the MCU enters Deep Software Standby 1 mode on execution of a WFI instruction.

While IWDTCSTPR.SLCSTP is 0 (counting continues), the setting to Deep software Standby 2 and 3 mode is ineffective. Even if LPMD is any of 0x9 or 0xA, the MCU enters Deep Software Standby 1 mode on execution of a WFI instruction.

Refer to [Table 10.9](#) for the detail.

While the voltage monitor  $n$  ( $n=1, 2$ ) reset is enabled (PVDnCR0.RI=1), the setting to Deep Software Standby mode 2, 3 is ineffective. Even if LPMD is any of 0x9 or 0xA the MCU enters Deep Software Standby 1 mode on execution of a WFI instruction. Refer to [Table 10.9](#) for the detail.

### 10.2.14 DPSBYCR : Deep Software Standby Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	IOKEEP P	—	SRKEEP EP	—	DCSS MODE	—	—
Value after reset:	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	DCSSMODE	DCDC Soft Start Mode 0: When the Deep Software Standby mode is canceled, the time required to recover is the standard time. 1: When the Deep Software Standby mode is canceled, the time required to recover is shortened.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	SRKEEP	Standby SRAM Retention 0: When entering the Software Standby mode or the Deep Software Standby mode 1, the contents of Standby SRAM are not kept. 1: When entering the Software Standby mode or the Deep Software Standby mode 1, the contents of Standby SRAM are kept.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	IOKEEP	I/O Port Rentention 0: When the Deep Software Standby mode is canceled, the I/O ports are in the reset state. 1: When the Deep Software Standby mode is canceled, the I/O ports are in the same state as in the Deep Software Standby mode.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The DPSBYCR register controls the Deep Software Standby mode.

DPSBYCR is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see [section 5, Resets](#).

#### DCSSMODE bit (DCDC Soft Start Mode)

The DCSSMODE bit specifies whether to shorten the time required to recover from the Deep Software Standby mode. For details, see [section 52, Electrical Characteristics](#).

**SRKEEP bit (Standby SRAM Retention)**

The SRKEEP bit specifies whether to keep the contents of Standby SRAM or not when entering the Software Standby mode or the Deep Software Standby mode 1.

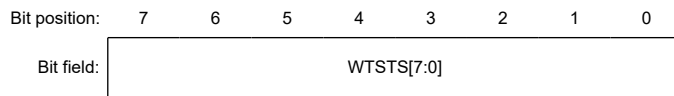
**IOKEEP bit (I/O Port Retention)**

In Deep Software Standby mode, I/O ports keep the same states as in the Software Standby mode. The IOKEEP bit specifies whether to reset the state of the I/O ports or not when the Deep Software Standby mode is canceled.

**10.2.15 DPSWCR : Deep Software Standby Wait Control Register**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA04



Value after reset: 0 0 0 0 1 0 1 1

Bit	Symbol	Function	R/W
7:0	WTSTS[7:0]	Deep Software Wait Standby Time Setting Bit 0x0B: Wait cycle for fast recovery 0x9A: Wait cycle for slow recovery*1 Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. Only Deep Software Standby mode 2 is supported.

The DPSWCR register is a register appointing waiting stabilization time when a Deep Software Standby mode is canceled by certain pins which are sources of external pin interrupts or peripheral interrupts.

During a waiting stabilization period appointed in this register, Deep Software Standby reset occurs, and this MCU is initialized.

The DPSWCR is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode. For details, see [section 5, Resets](#)

**WTSTS[7:0] bits (Deep Software Wait Standby Time Setting Bit)**

Set only 2 following values to WTSTS.

## 1. WTSTS = 0x0B:

The DSTBY recovery time is set to be fast. In this case, after returning from Deep Software Standby mode 2, it is necessary to be secured 700μs in the program before transition to next Software Standby mode or next Deep Software Standby mode.

For details of this fast recovery time, see [section 52, Electrical Characteristics](#)

## 2. WTSTS = 0x9A:

The DSTBY recovery time is set to be slow for Deep Software Standby mode 2. In this case, there is not the time constraint as the case of WTSTS = 0x0B.

For details of this slow recovery time, see [section 52, Electrical Characteristics](#)

Note: When you use Deep Software Standby mode 1 or Deep Software Standby mode 3, you should set 0x0B to WTSTS to avoid extra waiting time.

### 10.2.16 DPSIER0 : Deep Software Standby Interrupt Enable Register 0

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ7 E	DIRQ6 E	DIRQ5 E	DIRQ4 E	DIRQ3 E	DIRQ2 E	DIRQ1 E	DIRQ0 E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0E	IRQ0-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	DIRQ1E	IRQ1-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
2	DIRQ2E	IRQ2-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
3	DIRQ3E	IRQ3-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
4	DIRQ4E	IRQ4-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
5	DIRQ5E	IRQ5-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
6	DIRQ6E	IRQ6-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
7	DIRQ7E	IRQ7-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIER0 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see [section 5, Resets](#).

After the setting of DPSIER0 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR0 being set to 1. Therefore, DPSIFR0 should be cleared to 0 before entering Deep Software Standby mode.

Refer to [Table 10.4](#) to know the relation between this register and interrupt source for canceling Deep Software Standby mode.

### 10.2.17 DPSIER1 : Deep Software Standby Interrupt Enable Register 1

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ1 5E	DIRQ1 4E	DIRQ1 3E	DIRQ1 2E	DIRQ1 1E	DIRQ1 0E	DIRQ9 E	DIRQ8 E
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ8E	IRQ8-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	DIRQ9E	IRQ9-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
2	DIRQ10E	IRQ10-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
3	DIRQ11E	IRQ11-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
4	DIRQ12E	IRQ12-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
5	DIRQ13E	IRQ13-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
6	DIRQ14E	IRQ14-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
7	DIRQ15E	IRQ15-DS Pin Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIER1 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see [section 5, Resets](#).

After the setting of DPSIER1 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR1 being set to 1. Therefore, DPSIFR1 should be cleared to 0 before entering Deep Software Standby mode.

Refer to [Table 10.4](#) to know the relation between this register and interrupt source for canceling Deep Software Standby mode.

### 10.2.18 DPSIER2 : Deep Software Standby Interrupt Enable Register 2

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMI E	DRTC AIE	DRTC IE	DPVD 2IE	DPVD 1IE

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	DPVD1IE	PVD1 Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	DPVD2IE	PVD2 Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
2	DRTCIE	RTC Interval interrupt Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Bit	Symbol	Function	R/W
3	DRTCAIE	RTC Alarm interrupt Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
4	DNMIE	NMI Pin Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W <sup>1</sup>
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. 1 can be written only once. Once 1 is written to this bit, subsequent write accesses are disabled.

DPSIER2 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see [section 5, Resets](#).

After the setting of DPSIER2 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR2 being set to 1. Therefore, DPSIFR2 should be cleared to 0 before entering Deep Software Standby mode.

Refer to [Table 10.4](#) to know the relation between this register and interrupt source for canceling Deep Software Standby mode.

### 10.2.19 DPSIER3 : Deep Software Standby Interrupt Enable Register 3

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DVBATTADIE	—	DIWDTIE	—	DULPT1IE	DULPT0IE	—	DUSBFSIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DUSBFSIE	USBFS Suspend/Resume Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	DULPT0IE	ULPT0 Overflow Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
3	DULPT1IE	ULPT1 Overflow Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DIWDTIE	IWDT Underflow Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	DVBATTADIE	VBATT Tamper Detection Deep Software Standby Cancel Signal Enable 0: Canceling Deep Software Standby mode is disabled 1: Canceling Deep Software Standby mode is enabled	R/W

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIER3 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see [section 5, Resets](#).

After the setting of DPSIER3 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR3 being set to 1. Therefore, DPSIFR3 should be cleared to 0 before entering Deep Software Standby mode.



Refer to [Table 10.4](#) to know the relation between this register and interrupt source for canceling Deep Software Standby mode.

### 10.2.20 DPSIFR0 : Deep Software Standby Interrupt Flag Register 0

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ7 F	DIRQ6 F	DIRQ5 F	DIRQ4 F	DIRQ3 F	DIRQ2 F	DIRQ1 F	DIRQ0 F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0F	IRQ0-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
1	DIRQ1F	IRQ1-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
2	DIRQ2F	IRQ2-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
3	DIRQ3F	IRQ3-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
4	DIRQ4F	IRQ4-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
5	DIRQ5F	IRQ5-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
6	DIRQ6F	IRQ6-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
7	DIRQ7F	IRQ7-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a cancel request specified by DPSIEGR0 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not only in Deep Software Standby mode) or when the setting of DPSIER0 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR0 is cleared to 0x00.

To clear DPSIFR0 to 0x00 after modifying DPSIER0, wait for at least 6 PCLKB cycles, read DPSIFR0, and then write 0 to DPSIFR0. Six or more PCLKB cycles can be secured, for example, by reading DPSIER0.

DPSIFR0 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source.

For details, see [section 5, Resets](#).

#### DIRQnF flag (IRQn-DS Pin Deep Software Standby Cancel Flag) (n = 0 to 7)

The DIRQnF flag indicates that a cancel request by the IRQn-DS pin has been generated.

[Setting condition]

A cancel request by the IRQn-DS pin specified by DPSIEGR0 is generated.

[Clearing condition]

Writing 0 to each flag after 1 is read.

### 10.2.21 DPSIFR1 : Deep Software Standby Interrupt Flag Register 1

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA1C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ1 5F	DIRQ1 4F	DIRQ1 3F	DIRQ1 2F	DIRQ1 1F	DIRQ1 0F	DIRQ9 F	DIRQ8 F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ8F	IRQ8-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
1	DIRQ9F	IRQ9-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
2	DIRQ10F	IRQ10-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
3	DIRQ11F	IRQ11-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
4	DIRQ12F	IRQ12-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
5	DIRQ13F	IRQ13-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
6	DIRQ14F	IRQ14-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
7	DIRQ15F	IRQ15-DS Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a cancel request specified by DPSIEGR1 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not only in Deep Software Standby mode) or when the setting of DPSIER1 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR1 is cleared to 0x00.

To clear DPSIFR1 to 0x00 after modifying DPSIER1, wait for at least 6 PCLKB cycles, read DPSIFR1, and then write 0 to DPSIFR1. Six or more PCLKB cycles can be secured, for example, by reading DPSIER1.

DPSIFR1 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see [section 5, Resets](#).

#### DIRQnF flag (IRQn-DS Pin Deep Software Standby Cancel Flag) (n = 8 to 15)

The DIRQnF flag indicates that a cancel request by the IRQn-DS pin has been generated.

[Setting condition]

A cancel request by the IRQn-DS pin specified by DPSIEGR1 is generated.

[Clearing condition]

Writing 0 to each flag after 1 is read.

### 10.2.22 DPSIFR2 : Deep Software Standby Interrupt Flag Register 2

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA20

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMIF	DRTC AIF	DRTC IIF	DPVD 2IF	DPVD 1IF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPVD1IF	PVD1 Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
1	DPVD2IF	PVD2 Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
2	DRTC IIF	RTC Interval Interrupt Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
3	DRTC AIF	RTC Alarm Interrupt Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
4	DNMIF	NMI Pin Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when a cancel request specified by DPSIEGR2 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not only in Deep Software Standby mode) or when the setting of DPSIER2 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR2 is cleared to 0x00.

To clear DPSIFR2 to 0x00 after modifying DPSIER2, wait for at least 6 PCLKB cycles, read DPSIFR2, and then write 0 to DPSIFR2. Six or more PCLKB cycles can be secured, for example, by reading DPSIER2.

DPSIFR2 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see [section 5, Resets](#).

#### DPVDMIF flag (PVDm Deep Software Standby Cancel Flag) (m = 1 to 2)

The DPVDMIF flag indicates that a cancel request by the voltage monitor m signal has been generated.

[Setting condition]

A cancel request is generated by the voltage monitor m signal that is selected in DPSIEGR2.

[Clearing condition]

Writing 0 to each flag after 1 is read.

#### DRTC IIF flag (RTC Interval Interrupt Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the RTC interval interrupt signal has been generated.

[Setting condition]

A cancel request by the RTC interval interrupt signal is generated

[Clearing condition]

Writing 0 to each flag after 1 is read.

### DRTCAIF flag (RTC Alarm Interrupt Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the RTC alarm interrupt signal has been generated.

[Setting condition]

A cancel request by the RTC alarm interrupt signal is generated

[Clearing condition]

Writing 0 to each flag after 1 is read.

### DNMIF flag (NMI Pin Deep Software Standby Cancel Flag)

This flag indicates that a cancel request by the NMI pin has been generated.

[Setting condition]

A cancel request by the NMI pin specified by DPSIEGR2 is generated

[Clearing condition]

Writing 0 to each flag after 1 is read.

## 10.2.23 DPSIFR3 : Deep Software Standby Interrupt Flag Register 3

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DVBA TTADI F	—	DIWD TIF	—	DULP T1IF	DULP T0IF	—	DUSB FSIF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DUSBFSIF	USBFS Suspend/Resume Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
1	—	This bit is read as 0. The write value should be 0.	R/W
2	DULPT0IF	ULPT0 Overflow Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
3	DULPT1IF	ULPT1 Overflow Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DIWDTIF	IWDT Underflow Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>
6	—	This bit is read as 0. The write value should be 0.	R/W
7	DVBATTADIF	VBATT Tamper Detection Deep Software Standby Cancel Flag 0: The cancel request is not generated 1: The cancel request is generated	R/W <sup>1</sup>

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

Note 1. Only 0 can be written to clear flag. The flag must be cleared by writing 0 after 1 is read.

Each flag is set to 1 when the corresponding cancel request is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not only in Deep Software Standby mode) or when the setting of DPSIER3 is modified. Therefore, a transition to Deep Software Standby mode should be made after DPSIFR3 is cleared to 0x00.

To clear DPSIFR3 to 0x00 after modifying DPSIER3, wait for at least 6 PCLKB cycles, read DPSIFR3, and then write 0 to DPSIFR3. Six or more PCLKB cycles can be secured, for example, by reading DPSIER3.

DPSIFR3 is not initialized by the internal reset signal used as Deep Software Standby mode Canceling source. For details, see [section 5, Resets](#).

#### **DUSBFSIF flag (USBFS Suspend/Resume Deep Software Standby Cancel Flag)**

This flag indicates that a cancel request by the USBFS suspend/resume has been generated.

The DUSBFSIF flag is a flag for USBFS.

[Setting condition]

A cancel request by the USBFS suspend/resume is generated.

[Clearing condition]

Writing 0 to each bit after reading the bit as 1.

#### **DULPT0IF flag (ULPT0 Overflow Deep Software Standby Cancel Flag)**

This flag indicates that a cancel request by the ULPT0 overflow has been generated.

[Setting condition]

A cancel request by the ULPT0 is generated.

[Clearing condition]

Writing 0 to each bit after reading the bit as 1.

#### **DULPT1IF flag (ULPT1 Overflow Deep Software Standby Cancel Flag)**

This flag indicates that a cancel request by the ULPT1 overflow has been generated.

[Setting condition]

A cancel request by the ULPT1 is generated.

[Clearing condition]

Writing 0 to each bit after reading the bit as 1.

#### **DIWDTIF bit (IWDT Underflow Deep Software Standby Cancel Flag)**

This flag indicates that a cancel request by the IWDT underflow has been generated.

[Setting condition]

A cancel request by the IWDT is generated.

[Clearing condition]

Writing 0 to each bit after reading the bit as 1.

#### **DVBATTADIF bit (VBATT Tamper Detection Deep Software Standby Cancel Flag)**

This flag indicates that a cancel request by the VBATT tamper detection has been generated.

[Setting condition]

A cancel request by the VBATT tamper detection is generated.

[Clearing condition]

Writing 0 to each bit after reading the bit as 1.

### 10.2.24 DPSIEGR0 : Deep Software Standby Interrupt Edge Register 0

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA28

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ7 EG	DIRQ6 EG	DIRQ5 EG	DIRQ4 EG	DIRQ3 EG	DIRQ2 EG	DIRQ1 EG	DIRQ0 EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ0EG	IRQ0-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
1	DIRQ1EG	IRQ1-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
2	DIRQ2EG	IRQ2-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
3	DIRQ3EG	IRQ3-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
4	DIRQ4EG	IRQ4-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
5	DIRQ5EG	IRQ5-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
6	DIRQ6EG	IRQ6-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
7	DIRQ7EG	IRQ7-DS Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIEGR0 is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see [section 5, Resets](#).

### 10.2.25 DPSIEGR1 : Deep Software Standby Interrupt Edge Register 1

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA2C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DIRQ1 5EG	DIRQ1 4EG	DIRQ1 3EG	DIRQ1 2EG	DIRQ1 1EG	DIRQ1 0EG	DIRQ9 EG	DIRQ8 EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DIRQ8EG	IRQ8-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W

Bit	Symbol	Function	R/W
1	DIRQ9EG	IRQ9-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
2	DIRQ10EG	IRQ10-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
3	DIRQ11EG	IRQ11-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
4	DIRQ12EG	IRQ12-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
5	DIRQ13EG	IRQ13-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
6	DIRQ14EG	IRQ14-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W
7	DIRQ15EG	IRQ15-DS Pin Edge Select 0: A cancel request is generated at a falling edge. 1: A cancel request is generated at a rising edge.	R/W

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIEGR1 is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see [section 5, Resets](#).

### 10.2.26 DPSIEGR2 : Deep Software Standby Interrupt Edge Register 2

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA30

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DNMI EG	—	—	DPVD 2EG	DPVD 1EG
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPVD1EG	PVD1 Edge Select 0: A cancel request is generated when $VCC < V_{det1}$ (fall) is detected 1: A cancel request is generated when $VCC \geq V_{det1}$ (rise) is detected	R/W
1	DPVD2EG	PVD2 Edge Select 0: A cancel request is generated when $VCC < V_{det2}$ (fall) is detected 1: A cancel request is generated when $VCC \geq V_{det2}$ (rise) is detected	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DNMIEG	NMI Pin Edge Select 0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-4, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

DPSIEGR2 is not initialized by the internal reset signal that is the source to cancel the Deep Software Standby mode. For details, see [section 5, Resets](#).

### 10.2.27 SYOCDCCR : System Control OCD Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA38

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DBGEN	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	—	The read value is undefined. The write value should be 0.	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	DBGEN	Debugger Enable bit 0: On-chip debugger is disabled 1: On-chip debugger is enabled*1	R/W

Note: S-TYPE-5, P-TYPE-2

Note 1. The value of DBGEN can be set by software. However, it is not until the debugger authentication passes that the on-chip debugger function is valid. That is, even if the DBGEN is 1, the on-chip debugger function is invalid unless the debugger authentication passes. For detail, see [section 2, CPU](#).

This register is not controlled by any security attribute register (eg. LPMSAR, DPFSAR).

#### DBGEN bit (Debugger Enable bit)

The DBGEN bit enables the on-chip debugger function.

[Setting condition]

- Power-on reset is generated
- Voltage Monitor 0 reset is generated.
- Writing 1 to the bit.

[Clearing condition]

- Writing 0 to the bit.

Note: Certain restrictions apply in terms of the MCU states in which the DBGEN bit can be set to 1. For details, see [section 2.13.2. Restriction in OCD Emulator Connecting](#).

### 10.2.28 PLL1LDOCR : PLL1-LDO Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xB04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SKEEP	LDOSTP
Value after reset:	0	0	0	0	0	x	0	0

Bit	Symbol	Function	R/W
0	LDOSTP	LDO Stop 0: PLL1-LDO is enabled 1: PLL1-LDO is stopped	R/W
1	SKEEP	STBY Keep 0: PLL1-LDO is stopped during Software Standby mode. 1: PLL1-LDO state before Software Standby mode is retained during Software Standby mode.	R/W



Bit	Symbol	Function	R/W
2	—	The read value is undefined.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

It can be controlled the operation of the LDO dedicated for PLL1(PLL1-LDO) by using this register.

Writing to this register is prohibited in low-speed mode.

### LDOSTP bit (LDO Stop)

The PLL1-LDO state during Normal mode is controlled by this bit. After setting 1 to this bit, do not set 0 to this bit for at least 1  $\mu$ s. This bit can be set 0 to this bit only when 1 $\mu$ s has elapsed after setting 1 to this bit.

When this bit is changed from 1 to 0, it takes 25  $\mu$ s for the power supply to stabilize.

Allow the power supply to stabilize before operating the PLL1.

### SKEEP bit (STBY Keep)

The PLL1-LDO state during Software Standby mode is controlled by this bit. If the PLL1-LDO operation is stopped during Software Standby mode, the power consumption during Software Standby mode can be reduced.

## 10.2.29 PLL2LDOCR : PLL2-LDO Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xB08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SKEEP P	LDOSTP
Value after reset:	0	0	0	0	0	x	0	0

Bit	Symbol	Function	R/W
0	LDOSTP	LDO Stop 0: PLL2-LDO is enabled 1: PLL2-LDO is stopped	R/W
1	SKEEP	STBY Keep 0: PLL2-LDO is stopped during Software Standby mode. 1: PLL2-LDO state before Software Standby mode is retained during Software Standby mode.	R/W
2	—	The read value is undefined.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

It can be controlled the operation of the LDO dedicated for PLL2 (PLL2-LDO) by using this register.

Writing to this register is prohibited in low-speed mode.

### LDOSTP bit (LDO Stop)

The PLL2-LDO state during Normal mode is controlled by this bit. After setting 1 to this bit, do not set 0 to this bit for at least 1  $\mu$ s. This bit can be set 0 to this bit only when 1 $\mu$ s has elapsed after setting 1 to this bit.

When this bit is changed from 1 to 0, it takes 25  $\mu$ s for the power supply to stabilize.

Allow the power supply to stabilize before operating the PLL2.

### SKEEP bit (STBY Keep)

The PLL2-LDO state during Software Standby mode is controlled by this bit. If the PLL2-LDO operation is stopped during Software Standby mode, the power consumption during Software Standby mode can be reduced.

### 10.2.30 HOCOLDOCR : HOCO-LDO Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xB0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SKEE P	LDOS TP
Value after reset:	0	0	0	0	0	x	0	0

Bit	Symbol	Function	R/W
0	LDOSTP	LDO Stop 0: HOCO-LDO is enabled 1: HOCO-LDO is stopped	R/W
1	SKEEP	STBY Keep 0: HOCO-LDO is stopped during Software Standby mode. 1: HOCO-LDO state before Software Standby mode is retained during Software Standby mode.	R/W
2	—	The read value is undefined.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

It can be controlled the operation of the LDO dedicated for HOCO (HOCO-LDO) by using this register.

Writing to this register is prohibited in low-speed mode.

#### LDOSTP bit (LDO Stop)

The HOCO-LDO state during Normal mode is controlled by this bit. After setting 1 to this bit, do not set 0 to this bit for at least 2  $\mu$ s. This bit can be set 0 to this bit only when 2  $\mu$ s has elapsed after setting 1 to this bit.

When this bit is changed from 1 to 0, it takes 5  $\mu$ s for the power supply to stabilize.

Allow the power supply to stabilize before operating the HOCO.

#### SKEEP bit (STBY Keep)

The HOCO-LDO state during Software Standby mode is controlled by this bit. If the HOCO-LDO operation is stopped during Software Standby mode, the power consumption during Software Standby mode can be reduced.

## 10.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency changes when the SCKDIVCR and SCKDIVCR2 registers are set.

For information on module and clock associations, see [section 8.2.2. SCKDIVCR : System Clock Division Control Register](#) and [section 8.2.3. SCKDIVCR2 : System Clock Division Control Register 2](#).

## 10.4 Module-Stop Function

The module stop function can stop the clock supply set for each peripheral module.

When the MSTPmi bit (m = A to E, i = 31 to 0) in MSTPCRn (n = A to E) is set to 1, the specified module stops operating and enters the module-stop state, but the CPU continues to operate independently. Setting the MSTPmi bit to 0 cancels the module-stop state, allowing the module to resume operation at the end of the bus cycle.

After a reset is canceled, all modules other than the DMAC, DTC and SRAM1 modules are placed in the module-stop state. Do not access the module while the corresponding MSTPmi bit is 1. Additionally, do not set 1 to the MSTPmi bit while the corresponding module is accessed.

If the CPU clock is faster than 120 MHz, after the MSTPmi bit is changed, a wait time should be provided according to the flow shown in [Figure 10.2](#). It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt.

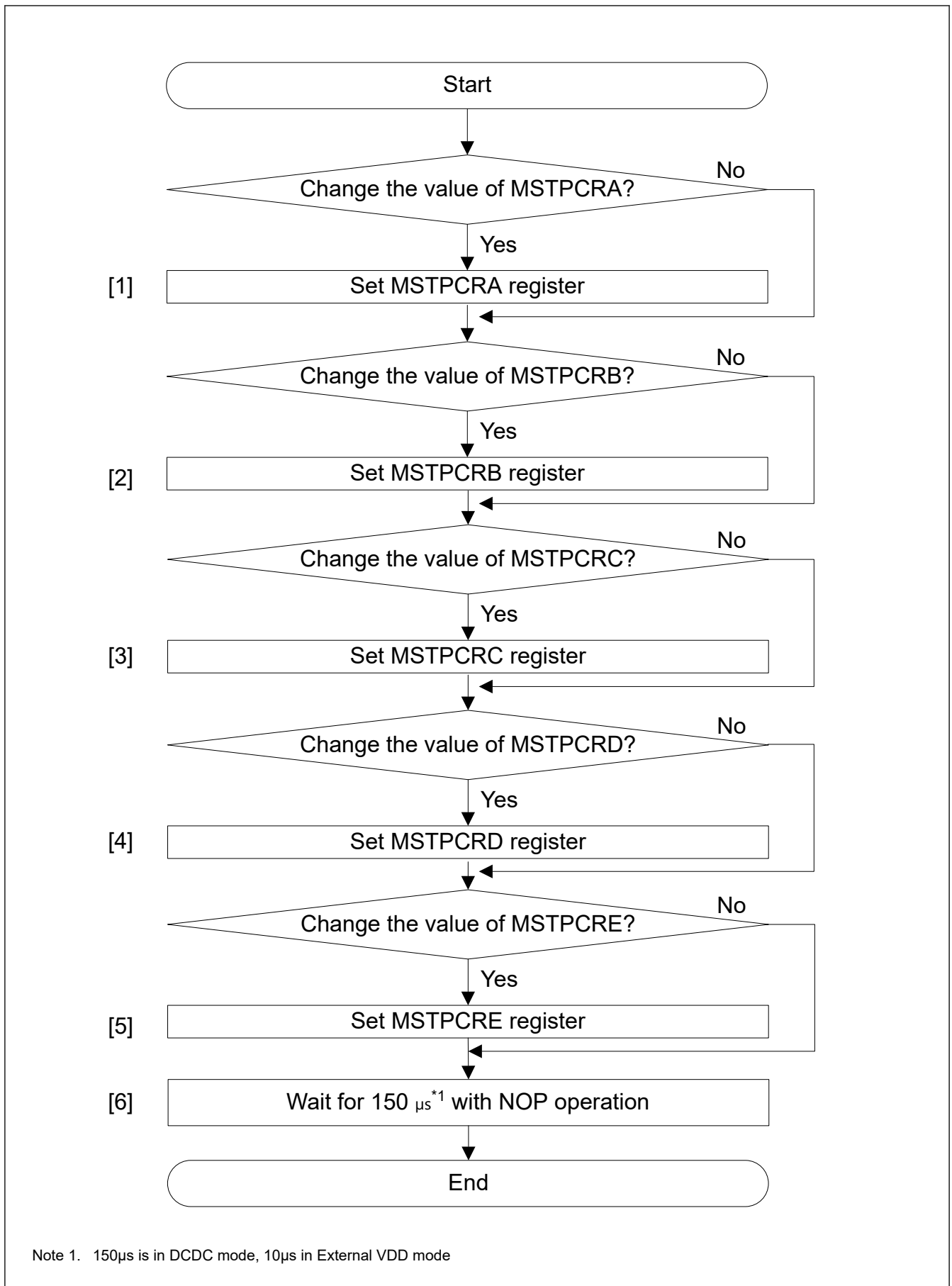


Figure 10.2 Setting flow of module stop control registers in case of CPUCLK is higher than 120MHz

## 10.5 Function for Lower Operating Power Consumption

By selecting an appropriate operating power consumption control mode according to the operating frequency, power consumption can be reduced in Normal mode and Processor low power mode.

### 10.5.1 Setting Operating Power Control Mode

Ensure the operating condition such as the frequency range is always within the specified range before and after switching the operating power control modes.

In External VDD mode, make sure to set OPCCR.OPCM is 00b.

This section provides example procedures for switching operating power control modes.

**Table 10.8 Available oscillators in each mode**

Mode	Oscillator					
	PLL1, PLL2	High-speed on-chip oscillator	Middle-speed on-chip oscillator	Low-speed on-chip oscillator	Main clock oscillator	Sub-clock oscillator
High-speed	Available	Available	Available	Available	Available	Available
Low-speed	Not Available	Available	Available	Available	Available	Available

#### (1) Switching from a higher power mode to a lower power mode

Example : From High-speed mode to Low-speed mode:

(Operation begins in High-speed mode)

1. Change the oscillator to what is used in Low-speed mode. Set the frequency of each clock lower than or equal to the maximum operating frequency in Low-speed mode.
2. Turn off the oscillator that is not required in Low-speed mode.
3. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
4. Set the OPCCR.OPCM[1:0] bits to 11b (Low-speed mode).
5. Confirm that OPCCR.OPCMTSF flag is 0 (indicates transition completed).

#### (2) Switching from a lower power mode to a higher power mode

Example : From Low-speed mode to High-speed mode

(Operation begins in Low-speed mode)

1. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
2. Set the OPCCR.OPCM[1:0] bits to 00b (High-speed mode).
3. Confirm that the OPCCR.OPCMTSF flag is 0 (indicates transition completed).
4. Turn on any required oscillator in High-speed mode.
5. Set the frequency of each clock lower than or equal to the maximum operating frequency for High-speed mode.

(Operation is now in High-speed mode)

## 10.6 Low Power Modes

### 10.6.1 CPU Sleep Mode

#### 10.6.1.1 Transition to CPU Sleep mode

When a WFI instruction is executed while SCR.SLEEPDEEP bit is 0, the MCU enters CPU Sleep mode. In this mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop. Available resets or interrupts in CPU Sleep mode cause the MCU to cancel CPU Sleep mode. All interrupt sources are available. If using an interrupt to cancel CPU Sleep mode, you must set the associated IELSRn register before executing a WFI instruction. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

Counting by IWDT stops when the MCU enters CPU Sleep mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in CPU Sleep mode or CPU Deep Sleep mode). Similarly, counting by IWDT stops when the MCU enters CPU Sleep mode while the IWDT is in register start mode and the SLCSTP bit in IWDCSTPR is 1 (IWDT stops in CPU Sleep mode or CPU Deep Sleep mode).

Counting by IWDT continues when the MCU enters CPU Sleep mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 0 (IWDT does not stop in CPU Sleep mode or CPU Deep Sleep mode). Similarly, counting by IWDT continues when the MCU enters CPU Sleep mode while the IWDT is in register start mode and the SLCSTP bit in IWDCSTPR is 0 (IWDT does not stop in CPU Sleep mode or CPU Deep Sleep mode).

Counting by WDT stops when the MCU enters CPU Sleep mode while the WDT is in auto start mode and the OFS0.WDTSTPCTL bit is 1 (WDT stops in CPU Sleep mode or CPU Deep Sleep mode). Similarly, counting by WDT stops when the MCU enters CPU Sleep mode while the WDT is in register start mode and the SLCSTP bit in WDCSTPR is 1 (WDT stops in CPU Sleep mode or CPU Deep Sleep mode).

Counting by WDT continues when the MCU enters CPU Sleep mode while the WDT is in auto start mode and the OFS0.WDTSTPCTL bit is 0 (WDT does not stop in CPU Sleep mode or CPU Deep Sleep mode). Similarly, counting by WDT continues when the MCU enters CPU Sleep mode while the WDT is in register start mode and the SLCSTP bit in WDCSTPR is 0 (WDT does not stop in CPU Sleep mode or CPU Deep Sleep mode).

WFI instruction and sleep-on-exit function for entering CPU Sleep mode purpose.

For the transition to CPU Sleep mode, there is a procedure according to the CPUCLK frequency. For details, see [section 10.7.10. Notes on transitioning to or canceling low power state](#).

### 10.6.1.2 Canceling CPU Sleep mode

CPU Sleep mode is canceled by:

- An interrupt
- A reset:
  - RES pin reset
  - Power-on reset
  - Voltage Monitor 0 reset
  - Independent watchdog timer reset
  - Watchdog timer reset
  - Voltage Monitor m reset (m = 1, 2)
  - Common memory error reset
  - Bus error reset

The operations are as follows:

1. Canceling by an interrupt  
When an available interrupt request is generated, CPU Sleep mode is canceled and the MCU starts the interrupt handling.
2. Canceling by RES pin reset  
When the RES pin is driven low, the MCU enters the reset state. Be sure to keep the RES pin low for the time period specified in [section 52, Electrical Characteristics](#) section 50, Electrical Characteristics. When the RES pin is driven high after the specified time period, the CPU starts the reset exception handling.
3. Canceling by IWDT reset  
CPU Sleep mode is canceled by an internal reset generated by an IWDT underflow and the MCU starts the reset exception handling. However, IWDT stops in CPU Sleep mode and an internal reset for canceling CPU Sleep mode is not generated in the following conditions:
  - OFS0.IWDTSTRT = 0 (auto start mode) and OFS0.IWDTSTPCTL = 1
  - OFS0.IWDTSTRT = 1 (register start mode) and OFS0.IWDTSTPCTL = 1
4. Canceling by WDT reset

CPU Sleep mode is canceled by an internal reset generated by a WDT underflow and the MCU starts the reset exception handling. However, WDT stops in CPU Sleep mode even when counting in Normal mode and an internal reset for canceling CPU Sleep mode is not generated in the following conditions:

- OFS0.WDTSTRT = 0 (auto start mode) and OFS0.WDTSTPCTL = 1
- OFS0.WDTSTRT = 1 (register start mode) and WDTCSSTPR.SLCSTP = 1

#### 5. Canceling by other resets available in CPU Sleep mode

CPU Sleep mode is canceled by other resets and the MCU starts the reset exception handling.

Note: For details on proper setting of the interrupts, see [section 13, Interrupt Controller Unit \(ICU\)](#)

## 10.6.2 CPU Deep Sleep mode

### 10.6.2.1 Transition to CPU Deep Sleep mode

When a WFI instruction is executed while SCR.SLEEPDEEP is set as 1, the MCU enters CPU Deep Sleep mode. In this mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop. However, accessing to TCM is not available and SysTick also stops in this mode. Available resets or wakeup enabled interrupts in CPU Deep Sleep mode cause the MCU to cancel CPU Deep Sleep mode. Not all interrupt sources are available, in detail user must also set WUPEN0 and WUPEN1 registers located in ICU to enable wakeup from CPU Deep Sleep mode. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

IWDT and WDT operation in CPU Deep Sleep mode is same with CPU Sleep mode.

For the transition to CPU Deep Sleep mode, there is a procedure according to the CPUCLK frequency. For details, see [section 10.7.10. Notes on transitioning to or canceling low power state](#).

### 10.6.2.2 Canceling CPU Deep Sleep mode

CPU Deep Sleep mode is canceled by:

- An available interrupt shown in [Table 10.4](#).
- A reset:
  - RES pin reset
  - Power-on reset
  - Voltage Monitor 0 reset
  - Independent watchdog timer reset
  - Watchdog timer reset
  - Voltage Monitor m reset (m = 1, 2)
  - Common memory error reset
  - Bus error reset

The operations are as follows:

#### 1. Canceling by an interrupt

When an available interrupt request (for available interrupt, See [Table 10.4](#)) is generated, each oscillator which was operating before the transition to CPU Deep Sleep mode restarts. After all these oscillators have been stabilized, the MCU returns from CPU Deep Sleep mode to Normal mode and starts interrupt handling. When canceling the CPU Deep Sleep mode by interrupt, each power domain returns to the same power state as the state before entering the Deep Sleep state

(For available interrupt, See [Table 10.4](#))

#### 2. Canceling by RES pin reset

When the RES pin is driven low, the MCU enters the reset state. Be sure to keep the RES pin low for the time period specified in [section 52, Electrical Characteristics](#). When the RES pin is driven high after the specified time period, the CPU starts the reset exception handling.

#### 3. Canceling by IWDT reset

CPU Deep Sleep mode is canceled by an internal reset generated by an IWDT underflow and the MCU starts the reset exception handling. However, IWDT stops in CPU Deep Sleep mode and an internal reset for canceling CPU Deep Sleep mode is not generated in the following conditions:

- OFS0.IWDTSTRT = 0 (auto start mode) and OFS0.IWDTSTPCTL = 1
- OFS0.IWDTSTRT = 1 (register start mode) and IWDCSTPR.SLCSTP = 1

#### 4. Canceling by WDT reset

CPU Deep Sleep mode is canceled by an internal reset generated by a WDT underflow and the MCU starts the reset exception handling. However, WDT stops in CPU Deep Sleep mode even when counting in Normal mode and an internal reset for canceling CPU Deep Sleep mode is not generated in the following conditions:

- OFS0.WDTSTRT = 0 (auto start mode) and OFS0.WDTSTPCTL = 1
- OFS0.WDTSTRT = 1 (register start mode) and WDCSTPR.SLCSTP = 1

#### 5. Canceling by other resets available in CPU Deep Sleep mode

CPU Deep Sleep mode is canceled by other resets and the MCU starts the reset exception handling.

Note: For details on proper setting of the interrupts, see [section 13, Interrupt Controller Unit \(ICU\)](#)

For CPU Deep Sleep mode cancellation, there is a procedure according to the CPUCLK frequency. For details, see [section 10.7.10. Notes on transitioning to or canceling low power state.](#)

### 10.6.3 Software Standby Mode

#### 10.6.3.1 Transition to Software Standby Mode

When a WFI instruction is executed while LPSCR.LPMD bit is 0x4 and CPU0.SCR.SLEEPDEEP bit is 1, the MCU enters Software Standby mode depending on LPSCR.LPMD setting. It must be set MOCO.CR.MCSTP to 0 (MOCO is operated) before executed the WFI instruction to entry this mode.

In this mode, the CPU, most of the on-chip peripheral functions and most of the oscillators stop<sup>\*1</sup>. However, the contents of the CPU internal registers and SRAM data, the states of on-chip peripheral functions and the I/O ports are retained. Software Standby mode allows significant reduction in power consumption because most of the oscillators stops<sup>\*1</sup> in this mode. [Table 10.3](#) shows the status of each on-chip peripheral functions and oscillators. Available resets or interrupts in Software Standby mode make the MCU to return to Normal mode. See [Table 10.4](#) for available interrupt sources for each mode. In case of using an interrupt to return to Normal mode, corresponding IELSRn register must be set before executing a WFI instruction. For the detail, see [section 13, Interrupt Controller Unit \(ICU\)](#).

Note 1. The state of the oscillators depends on the setting of the control register for each oscillator. For details, refer to the chapter of clock generation circuit.

Clear DMAST.DMST bit and DTCST.DTCST bit to 0 before executing WFI instruction.

Counting by IWDT stops when the MCU enters Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 1 (IWDT stops in Software Standby mode). Similarly, counting by IWDT stops when the MCU enters Software Standby mode while the IWDT is in register start mode and the SLCSTP bit in IWDCSTPR is 1 (IWDT stops in Software Standby mode).

Counting by IWDT continues when the MCU enters Software Standby mode while the IWDT is in auto start mode and the OFS0.IWDTSTPCTL bit is 0 (IWDT does not stop in Software Standby mode). Similarly, counting by IWDT continues when the MCU enters Software Standby mode while the IWDT is in register start mode and the SLCSTP bit in IWDCSTPR is 0 (IWDT does not stop in Software Standby mode).

WDT stops counting when the MCU enters Software Standby mode because the PCLK stops.

Do not enter Software Standby mode while the flash memory is in programming or erasing procedure. In order to enter Software Standby mode, execute a WFI instruction after programming or erasing procedure completes.

In External VDD mode, do not enter Software Standby Mode.

[Table 10.9](#) shows the setting of the related control bits and the modes to enter after executing WFI instruction.

**Table 10.9** Setting of the bits what affect the mode after executing WFI instruction

		Setting mode of LPSCR.LPMD bit and CPU.SCR.SLEEPDEEP bit			
		Software Standby	Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
		(LPSCR.LPMD = 0x4, CPU0.SCR.SLEEPDEEP = 1)	(LPSCR.LPMD = 0x8, CPU0.SCR.SLEEPDEEP = 1)	(LPSCR.LPMD = 0x9, CPU0.SCR.SLEEPDEEP = 1)	(LPSCR.LPMD = 0xA, CPU0.SCR.SLEEPDEEP = 1)
FENTRYR.FENTRYC FENTRYR.FENTRYD	0	Software Standby	Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
	1	CPU Deep Sleep mode and Keep System Active	CPU Deep Sleep mode and Keep System Active	CPU Deep Sleep mode and Keep System Active	CPU Deep Sleep mode and Keep System Active
OFS0.IWDTSTPCTL (auto-start mode) IWDCSTPR.SLCSTP (register start mode)	0	Software Standby	Deep Software Standby 1	Deep Software Standby 1	Deep Software Standby 1
	1			Deep Software Standby 2	Deep Software Standby 3
PVD1CR0.RI PVD2CR0.RI	0	Software Standby	Deep Software Standby 1	Deep Software Standby 2	Deep Software Standby 3
	1			Deep Software Standby 1	Deep Software Standby 1

Note: When multiple mode transition conditions conflict, mode transition is performed according to the following priority order.  
 CPU Deep Sleep mode and Keep System Active > Software Standby > Deep Software Standby 1 > Deep Software Standby 2 > Deep Software Standby 3.

For the transition to Software Standby mode, there is a procedure according to the CPUCLK frequency. For details, see [section 10.7.10. Notes on transitioning to or canceling low power state.](#)

### 10.6.3.2 Canceling Software Standby Mode

Software Standby mode is canceled by:

- An available interrupt shown in [Table 10.4](#)
- A reset:
  - RES pin reset
  - Power-on reset
  - Voltage monitor reset
  - Reset caused by an IWDT underflow

On exiting Software Standby mode, the oscillators that operate before the transition to the mode restart. After all the oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode. See [section 13.2.15. WUPEN0 : Wake Up Interrupt Enable Register 0](#), [section 13.2.16. WUPEN1 : Wake Up interrupt enable register 1](#) for information on how to wake up the MCU from Software Standby mode.

You can cancel Software Standby mode in any of the following ways:

1. Canceling by an interrupt
 

When an available interrupt request (see [Table 10.4](#)) is generated, an oscillator that operates before the transition to Software Standby mode restarts. After all the oscillators are stabilized, the MCU returns to Normal mode from Software Standby mode and starts the interrupt handling.

When canceling the software standby mode by interrupt, each power domain returns to the same power state as the state before entering the software standby state.
2. Canceling by a RES pin reset
 

When the RES pin is driven low, the MCU enters the reset state, and the oscillators whose default status is operating, start the oscillation. Be sure to keep the RES pin low for the time period specified in [section 52, Electrical Characteristics](#). When the RES pin is driven high after the specified time period, the CPU starts the reset exception handling.
3. Canceling by a power-on reset



Software Standby mode is canceled by a power-on reset and the MCU starts the reset exception handling.

4. Canceling by a voltage monitor reset

Software Standby mode is canceled by a voltage monitor reset from the voltage detection circuit and the MCU starts the reset exception handling.

5. Canceling by IWDTC reset

Software Standby mode is canceled by an internal reset generated by an IWDTC underflow and the MCU starts the reset exception handling. However, IWDTC stops in Software Standby mode and an internal reset for canceling Software Standby mode is not generated in the following condition:

- OFS0.IWDTCSTRT = 0 (auto start mode) and OFS0.IWDTCSTPCTL = 1
- OFS0.IWDTCSTRT = 1 (register start mode) and IWDTCSTPR.SLCSTP = 1

For Software Standby mode cancellation, there is a procedure according to the CPUCLK frequency. For details, see [section 10.7.10. Notes on transitioning to or canceling low power state](#)

### 10.6.3.3 Example of Software Standby Mode Application

Figure 10.3 shows an example of entry to Software Standby mode on detection of a falling edge of the IRQn pin, and exit from Software Standby mode by a rising edge of the IRQn pin.

In this example, an IRQn pin interrupt is accepted with the IRQCRi.IRQMD[1:0] (i = 0 to 15) bits of the ICU set to 00b (falling edge) in Normal mode, and the IRQCRi.IRQMD[1:0] bits are set to 01b (rising edge). Then, if MOCO is not operated, it must set MOCO.CR.MCSTP to 0 (MOCO is operated). After that, the LPSCR.LPMD bit is set to 0x4 and CPU.SCR.SLEEPDEEP bit is set to 1, and then a WFI instruction is executed. As a result, entry to Software Standby mode completes and exit from Software Standby mode is initiated by a rising edge of the IRQn pin.

Setting the ICU is also required to exit Software Standby mode. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#). The oscillation stabilization time in Figure 10.3 is specified in [section 52, Electrical Characteristics](#).

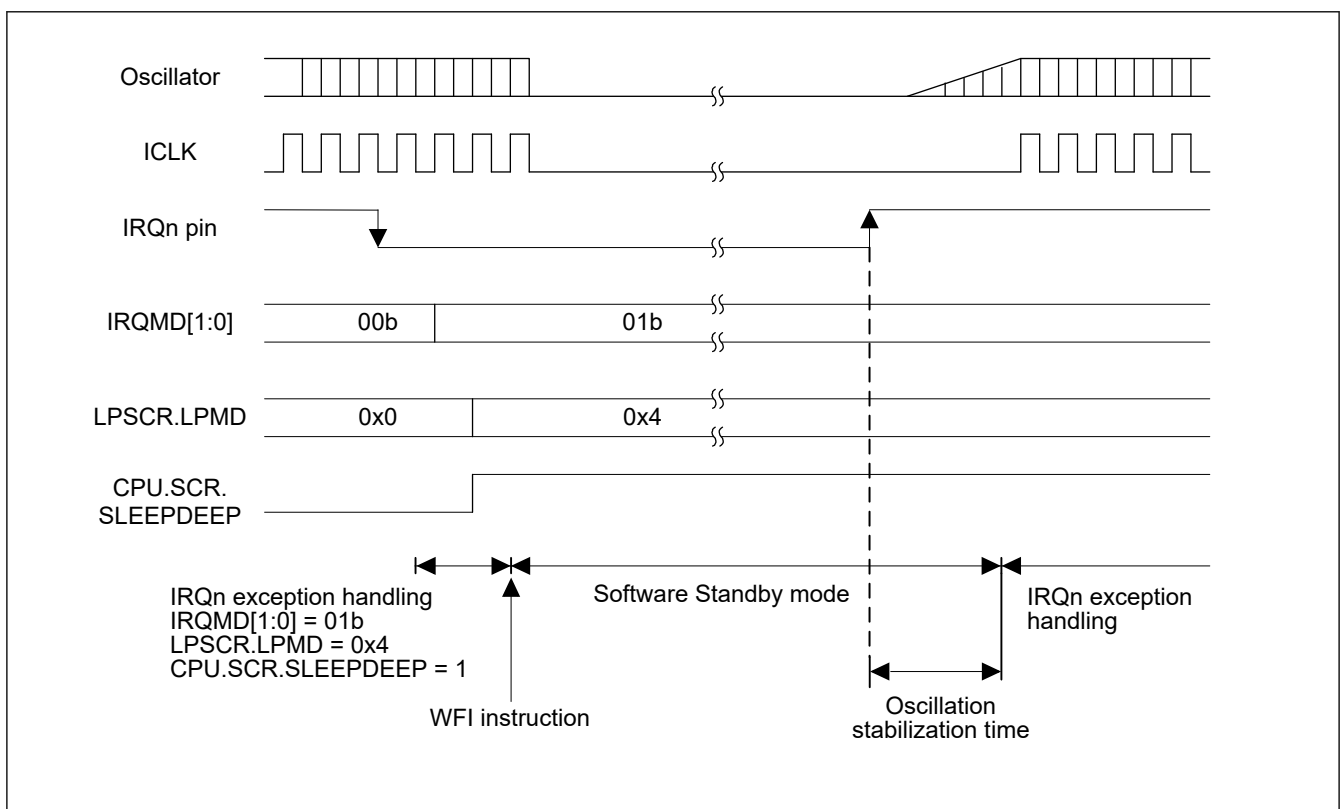


Figure 10.3 Example of Software Standby mode application

## 10.6.4 Deep Software Standby Mode

### 10.6.4.1 Transition to Deep Software Standby Mode

When a WFI instruction is executed while LPSCR.LPMD bit is 0x8, 0x9, 0xA and CPU.SCR.SLEEPDEEP bit is 1, the MCU enters Deep Software Standby mode 1 or 2 or 3 depending on LPSCR.LPMD setting. It must be set MOCO.CR.MCSTP to 0 (MOCO is operated) before executed the WFI instruction to entry this mode. See [Table 10.9](#) for the setting of the related control bits.

In these mode, the CPU, on-chip peripheral functions, SRAM, and oscillators are almost stopped as shown in [Table 10.3](#). Furthermore, since the internal power supply to these inactive modules is stopped, power consumption is remarkably reduced. The contents of all the registers of the CPU and specific internal peripheral modules become undefined. [Table 10.3](#) shows the status of each on-chip peripheral functions and oscillators.

Data in the standby SRAM are preserved in Deep Software Standby mode 1. When OFS1(\_SEC).PVDAS = 0 and OFS1(\_SEC).PVDLPSEL = 0, the low-power-consumption function of PVD0 is enabled, so power consumption is further reduced. In Deep Software Standby mode 2, the internal supply of power to the standby SRAM, the USB resume detecting unit and IWDT are cut off, reducing power consumption. Data in the standby SRAM become undefined at this time. When OFS1(\_SEC).PVDAS = 0 and OFS1(\_SEC).PVDLPSEL = 0, the low-power-consumption function of PVD0 is enabled, so power consumption is further reduced. In Deep Software Standby mode 3, the PVD is also stopped. For details, see [section 52, Electrical Characteristics](#).

Clear DMAST.DMST bit and DTCST.DTCST bit to 0 before executing WFI instruction.

When the MCU enters Deep Software Standby mode 2 or 3 while the IWDT is in auto-start mode and the OFS0.IWDTSTPCTL bit is 1, power supply to the LOCO and the IWDT is cut off. Counting by the IWDT also stops.

When the IWDT is in auto-start mode and the OFS0.IWDTSTPCTL bit is 0, the MCU enters Deep Software Standby mode 1 instead of Deep Software Standby mode 2 or 3, regardless of the setting in the LPSCR.LPMD.

When the IWDT is in register start mode and the IWDTCSTPR.SLCSTP is 0, the MCU enters Deep Software Standby mode 1 instead of Deep Software Standby mode 2 or 3, regardless of the setting in the LPSCR.LPMD.

Do not enter Deep Software Standby mode while the flash memory is in programming or erasing procedure. In order to enter Deep Software Standby mode, execute a WFI instruction after programming or erasing procedure completes.

In External VDD mode, do not enter Deep Software Standby mode.

When PVDnCR0.RI (n = 1, 2) = 1 (selecting the voltage monitor n reset), the MCU enters Deep Software Standby mode 1 instead of Deep Software Standby mode 2 or 3, regardless of the setting in the LPSCR.LPMD. The I/O port states are same as Software Standby mode.

For the transition to Deep Software Standby mode, there is a procedure according to the CPUCLK frequency. For details, see [section 10.7.10. Notes on transitioning to or canceling low power state](#)

### 10.6.4.2 Canceling Deep Software Standby Mode

Deep Software Standby mode is canceled by:

- An available interrupt shown in [Table 10.4](#)
- A reset:
  - RES pin reset
  - Power-on reset
  - Voltage monitor reset
  - Reset caused by an IWDT underflow

#### 1. Canceling by an interrupt

Canceling by interrupts is controlled by DPSIERn (n = 0 to 3) and DPSIFRn (n = 0 to 3). When a Deep Software Standby Canceling interrupt is generated, the corresponding flag in DPSIFRn is set to 1. If the interrupt is enabled in DPSIERn, Deep Software Standby mode is canceled. Rising edge or falling edge can be selected by DPSIEGRn (n = 0 to 2). The interrupts for which an edge can be selected are the NMI, IRQ0-DS to IRQ15-DS, voltage monitor n (n = 1, 2) interrupts. When a Deep Software Standby mode canceling request occurs, the internal power is supplied and MOCO starts oscillating, and then the internal reset (Deep Software Standby reset) is generated for the entire MCU.

The stable MOCO clock is then supplied to the entire MCU and Deep Software Standby reset is canceled. And the MCU starts reset exception handling.

When Deep Software Standby mode is canceled by an external interrupt pin or internal interrupt signal, the RSTSR0.DPSRSTF flag is set to 1.

2. Canceling by RES pin reset

When the RES pin is driven low, the MCU cancels Deep Software Standby mode and enters the reset state. Make sure to keep RES pin low for the time period specified in the chapter of Electrical Specification. When RES pin is driven high after the specified time period, the CPU starts the reset exception handling.

3. Canceling by a power-on reset

Deep Software Standby mode is canceled by a power-on reset and the MCU starts the reset exception handling.

4. Canceling by a voltage monitor 0 reset

Deep Software Standby mode is canceled by a voltage monitor 0 reset from the voltage detection circuit and the MCU starts the reset exception handling.

5. Canceling by a voltage monitor n reset (n = 1, 2)

Deep Software Standby mode 1 is canceled by a voltage monitor n reset (n= 1,2) from the voltage detection circuit and the MCU starts the reset exception handling.

6. Canceling by a IWDTC reset

Deep Software Standby mode 1 is canceled by a IWDTC reset (an internal reset generated by IWDTC underflow) and the MCU starts the reset exception handling.

However, IWDTC stops in Deep Software Standby mode 1 and a IWDTC reset for Canceling Deep Software Standby mode 1 is not generated in the following conditions:

- OFS0.IWDTCSTRT = 0 (auto start mode) and OFS0.IWDTCSTPCTL = 1
- OFS0.IWDTCSTRT = 1 (register start mode) and IWDTCSTPR.SLCSTP = 1

### 10.6.4.3 Pin States when Deep Software Standby mode is Canceled

In Deep Software Standby mode, the I/O ports retain the same states as before transition to the Deep Software Standby mode. The MCU is initialized by an internal reset generated when Deep Software Standby mode is canceled. Upon cancellation of Deep Software Standby mode, the reset exception handling starts immediately.

Whether to initialize the I/O ports or to retain the I/O port states as before transition to the Deep Software Standby mode can be selected by the DPSBYCR.IOKEEP bit.

- When the DPSBYCR.IOKEEP bit = 0

I/O ports are initialized by an internal reset generated when Deep Software Standby mode is canceled.

- When the DPSBYCR.IOKEEP bit = 1

Although the MCU is initialized by an internal reset generated when Deep Software Standby mode is canceled, I/O ports retain their states as before transition to the Deep Software Standby mode regardless of the MCU internal state. At this time, the I/O port states remain unchanged as before transition to the Deep Software Standby mode even if settings of I/O ports or peripheral modules are made. Then, the retained I/O port states are released by clearing the DPSBYCR.IOKEEP bit to 0, and the MCU operates according to the internal state.

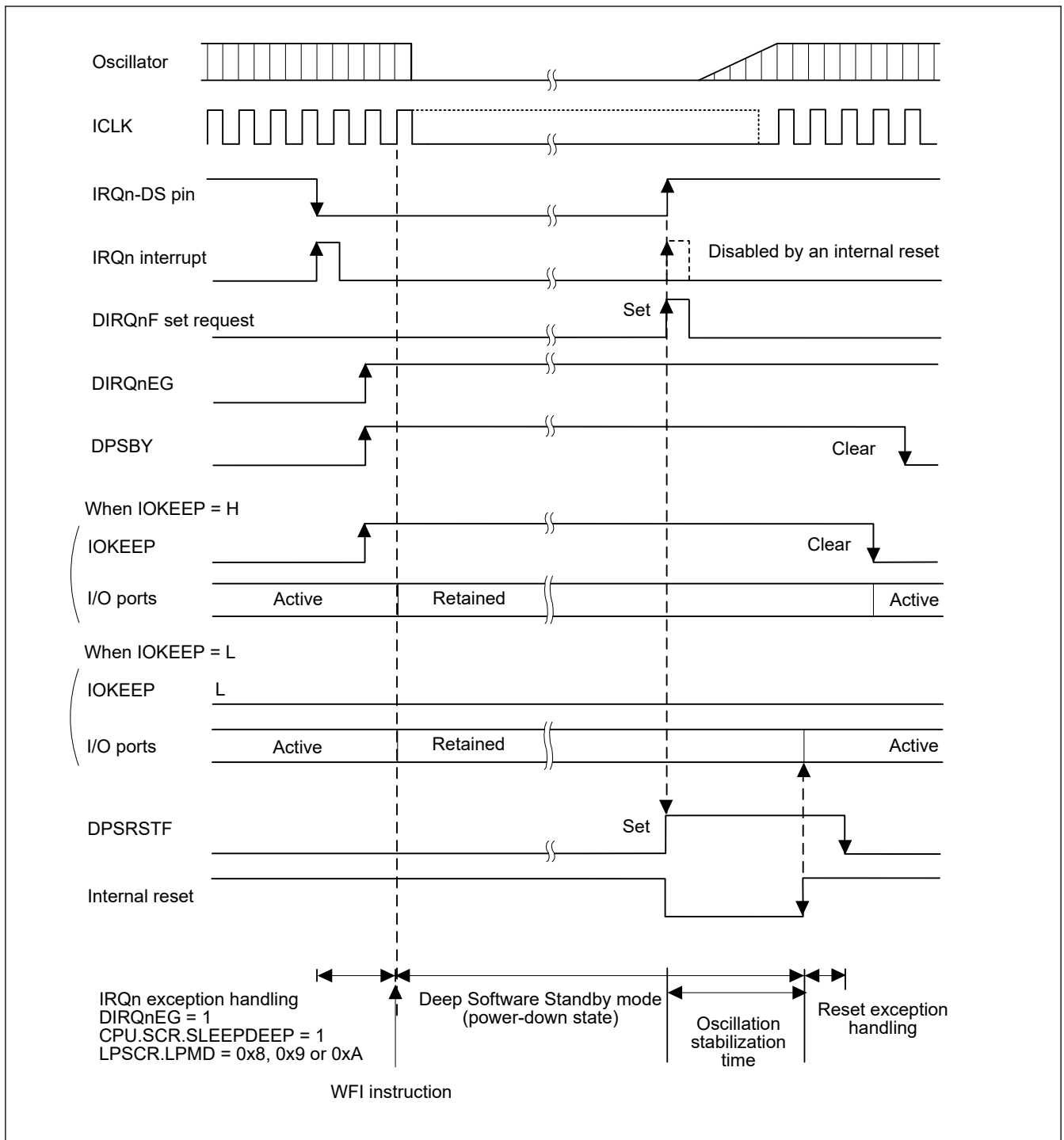
The DPSBYCR.IOKEEP bit is not initialized by the internal reset signal (Deep Software Standby reset signal) that is the source to cancel the Deep Software Standby mode.

### 10.6.4.4 Example of Deep Software Standby Mode Application

#### (1) Entering and exiting Deep Software Standby mode

Figure 10.4 shows an example where a transition to Deep Software Standby mode is made at the falling edge of the IRQn-DS pin, and exiting Deep Software Standby mode is made at the rising edge of the IRQn-DS pin. In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 00b (falling edge). If MOCO is not operated, it must set MOCO.CR.MCSTP to 0 (MOCO is operated).

Then, after the DPSIEGRy.DIRQnEG (y = 0 or 1, n = 0 to 15) bit is set to 1 (rising edge) and the LPSCR.LPMD bit is set to 0x8 or 0x9 or 0xA and CPU.SCR.SLEEPDEEP bit is set to 1, the WFI instruction is executed. Thus a transition to Deep Software Standby mode is made. After that, exiting Deep Software Standby mode is made at the rising edge of the IRQn-DS pin.



**Figure 10.4 Example of Deep Software Standby mode application**

### 10.6.4.5 Usage Flow for Deep Software Standby Mode

Figure 10.5 shows an example flow for using Deep Software Standby mode.

In this example, the RSTSR0.DPSRSTF flag of the reset function is read after the reset exception handling to determine whether a reset was generated by the RES pin or by the cancellation of Deep Software Standby mode.

For a reset by the RES pin, the MCU transitions to Deep Software Standby mode after the required register settings are made.

For a reset by cancellation of Deep Software Standby mode, the DPSBYCR.IOKEEP bit is cleared to 0 after the I/O port settings are made.

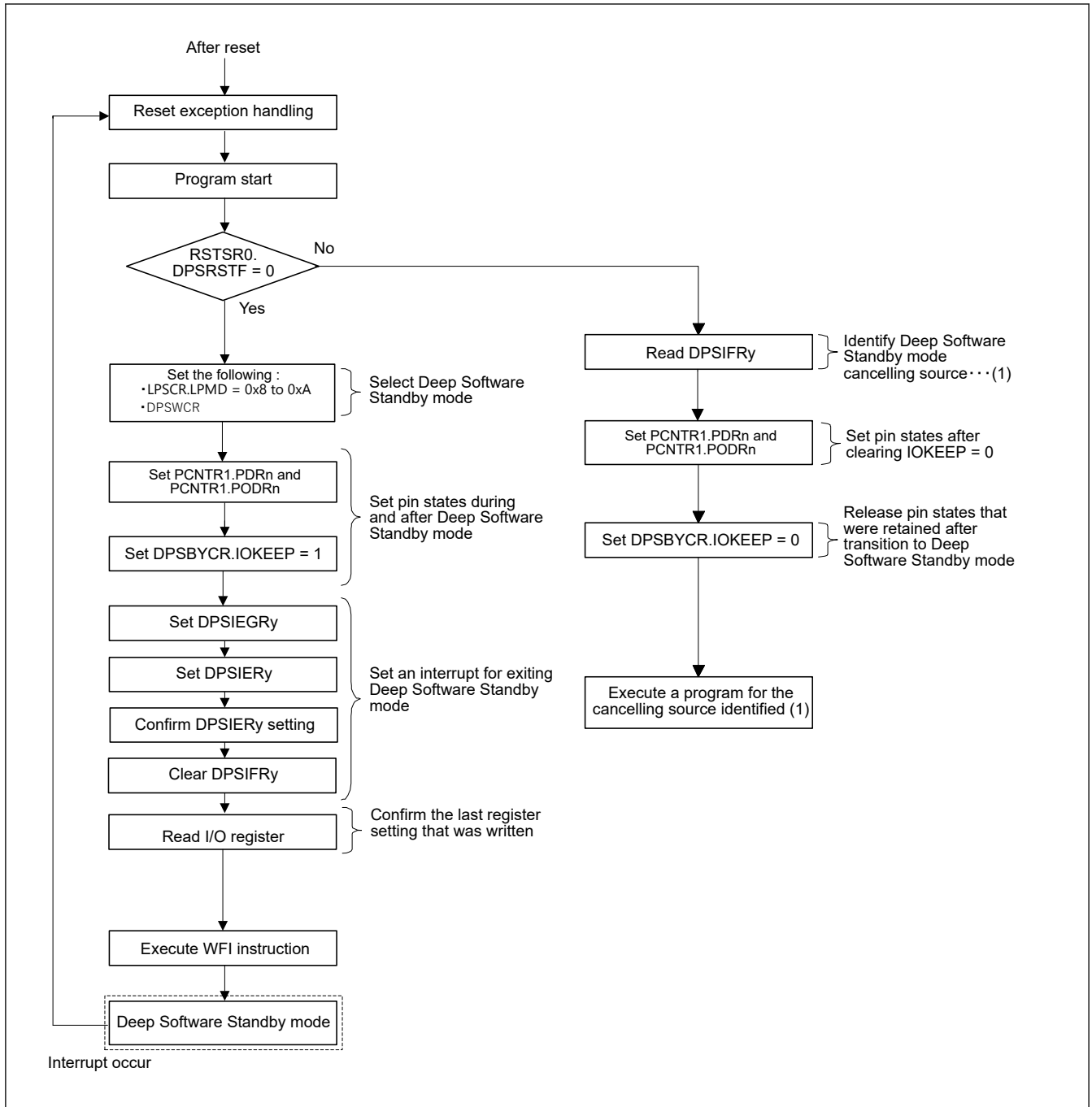


Figure 10.5 Example flow for using Deep Software Standby mode

## 10.7 Usage Notes

### 10.7.1 Register Access

(1) Invalid register write accesses during specific modes or transitions:

Do not write to registers under any of the conditions listed in this section.

[Registers]

- All registers with a peripheral name of “SYSTEM”.

[Conditions]

- OPCCR.OPCMTSF = 1 (during transition of operating power control mode)

- FENTRYR.FENTRYi = 1 (i = 0 to 3) or FENTRYR.FENTRYD = 1 (Flash P/E mode, dataFlash P/E mode)
- During the time period from executing a WFI instruction to returning to Normal mode

### (2) Valid setting for the clock-related registers

Table 10.10 and Table 10.11 shows the valid setting of the clock related registers in each operating power control mode. Do not write any value other than the valid setting. Also, each register has some prohibited settings under certain condition other than the operating power control modes. See section 8, [Clock Generation Circuit](#) for another condition of each register.

**Table 10.10 Valid setting of the clock related registers(1)**

Mode	Valid settings		
	SCKSCR.CKSEL[2:0] CKOCR.CKOSEL[2:0]	PLLCR.PLLSTP	PLL2CR.PLL2STP
High-speed	000b (HOCO) 001b (MOCO) 010b (LOCO) <sup>*2</sup> 011b (MOSC) 100b (SOSC) 101b (PLL1P) <sup>*1</sup>	0 (operating) 1 (stop)	0 (operating) 1 (stop)
Low-speed	000b (HOCO) 001b (MOCO) 010b (LOCO) <sup>*2</sup> 011b (MOSC) 100b (SOSC)	1 (stop)	1 (stop)

Note 1. SCKSCR.CKSEL[2:0] only

Note 2. CKOCR.CKOSEL[2:0] only

**Table 10.11 Valid setting of the clock related registers (2)**

Operating oscillator	Valid settings
	OPCCR.OPCM[1:0]
PLL1, PLL2	00b
High-speed on-chip oscillator	00b, 11b
Middle-speed on-chip oscillator	
Main clock oscillator	
Low-speed on-chip oscillator	
Sub-clock oscillator	

### (3) Invalid register write accesses by the DTC or DMAC

Do not write to registers listed in this section by the DTC or DMAC.

[Registers]

- MSTPCRA
- MSTPCRB
- MSTPCRC
- MSTPCRD
- MSTPCRE

### (4) Invalid write access when PRCR.PRC1 is 0

Write access to the registers listed below is invalid when PRCR.PRC1 bit is 0.

[Registers]

- OPCCR
- PDRAMSCRn
- SBYCR

- SSCR1
- LPSCR
- DPSBYCR
- DPSIER<sub>n</sub>
- DPSIFR<sub>n</sub>
- DPSIEGR<sub>n</sub>
- PLL1LDOSCR
- PLL2LDOSCR
- HOCOLDOSCR
- LVOCR

#### (5) Invalid write access when PRCR.PRC4 is 0

Write access to the registers listed below is invalid when PRCR.PRC4 bit is 0.

[Registers]

- LPMSAR
- PGCSAR
- DPFSAR
- RSCSAR

### 10.7.2 I/O Port States

I/O port states in Software Standby mode, Deep Software Standby mode are as same as before entering the modes. Therefore, the supply current is not reduced while output signals are held high.

### 10.7.3 Module-Stop State of DMAC and DTC

Before writing 1 to MSTPCRA.MSTPA22, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0. For details, see [section 16, DMA Controller \(DMAC\)](#) and [section 17, Data Transfer Controller \(DTC\)](#)

### 10.7.4 Internal Interrupt sources

Interrupts do not operate in the module-stop state. If setting module-stop bit while an interrupt request is generated, a CPU interrupt source or a DMAC or DTC startup source cannot be cleared. For this reason, make sure to disable the corresponding interrupts before setting the module-stop bits.

### 10.7.5 Input Buffer Control by DIRQnE Bit

Setting the DPSIER<sub>y</sub>.DIRQnE ( $y = 0$  or  $1$ ,  $n = 0$  to  $15$ ) bit to 1 enables the input buffer of the IRQ0-DS to IRQ15-DS pins. Note that, although inputs to these pins are sent to the DPSIFR<sub>y</sub>.DIRQnF ( $y = 0$  or  $1$ ,  $n = 0$  to  $15$ ) bits, they are not sent to the interrupt controller, peripheral modules, and I/O ports.

### 10.7.6 Transition to Low Power Modes

Since the MCU does not support wakeup by event, do not enter low power modes (CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode, or Deep Software Standby mode) by executing a WFE instruction.

### 10.7.7 Timing of WFI Instructions

It is possible for the WFI instruction to be executed before I/O register writes are complete, in which case operation may not be as intended. This can happen if the WFI is placed immediately after a write to a I/O register. To avoid this problem it is advisable to read back the register that was written to confirm that the write has completed.

### 10.7.8 Writing WDT/IWDT registers by DMAC or DTC

Do not write registers in WDT or IWDT by DMAC or DTC while WDT or IWDT stops by entering Processor low power mode.

### 10.7.9 Module stop bit write timing

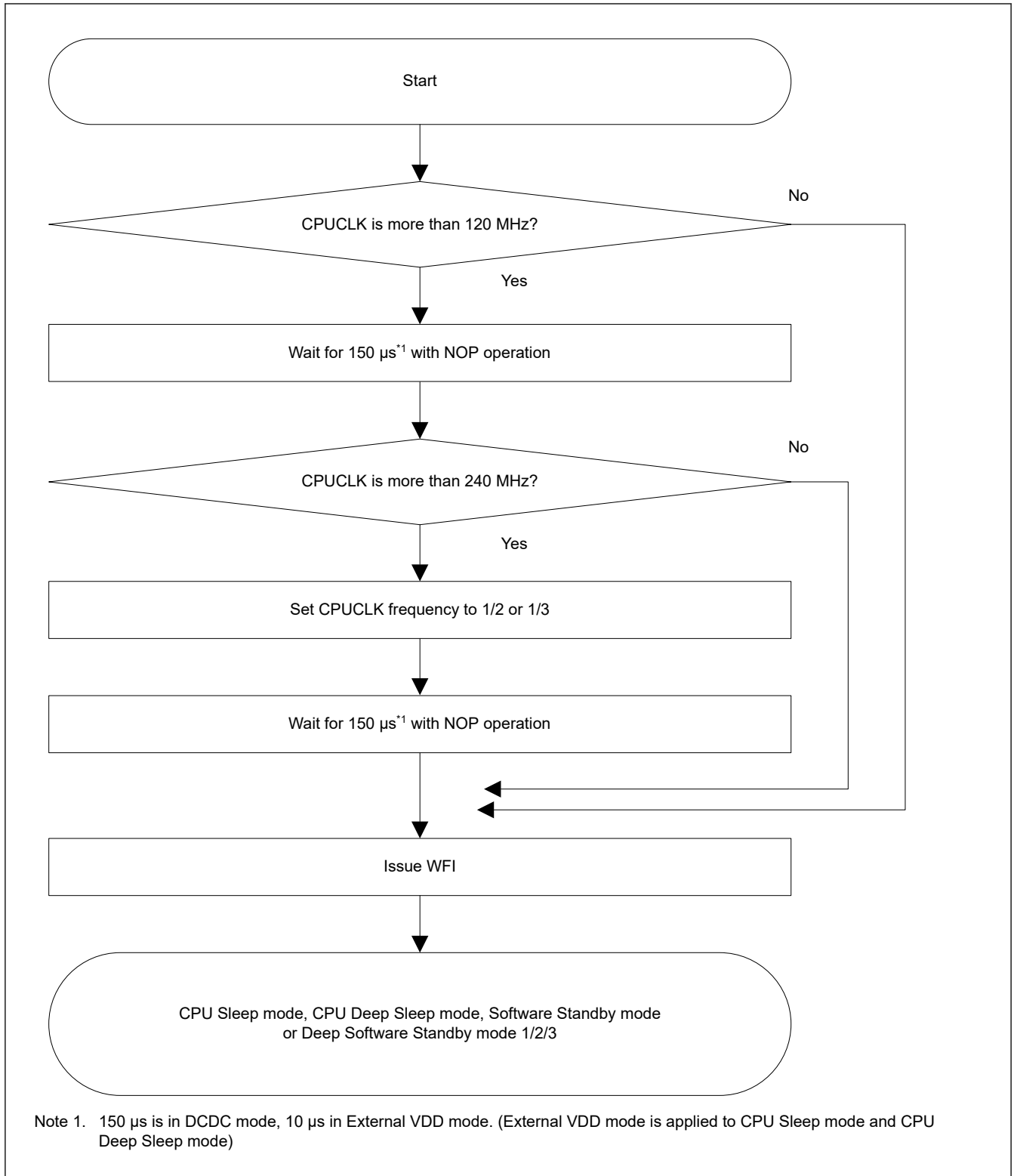
It is possible that the access to I/O register may be executed before the corresponding module stop bit write is completed. In this case, the access to I/O register may not proceed as intended. To avoid this issue, before accessing I/O register, read back the module stop bit that was written to confirm that the write completed.

### 10.7.10 Notes on transitioning to or canceling low power state

Transitioning to/returning from the low power state requires procedures according to the CPUCLK frequency.

These flows are shown in [Figure 10.6](#), [Figure 10.7](#) and [Figure 10.8](#). It is recommended to use software to measure the wait time. Be sure to consider the worst-case conditions to ensure that the required wait time elapses. If an interrupt is unavoidably generated during the wait time, retry the measurement after return from the interrupt





**Figure 10.6 Additional flow for the transition to CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode and Deep Software Standby mode 1/2/3 according to CPUCLK frequency**

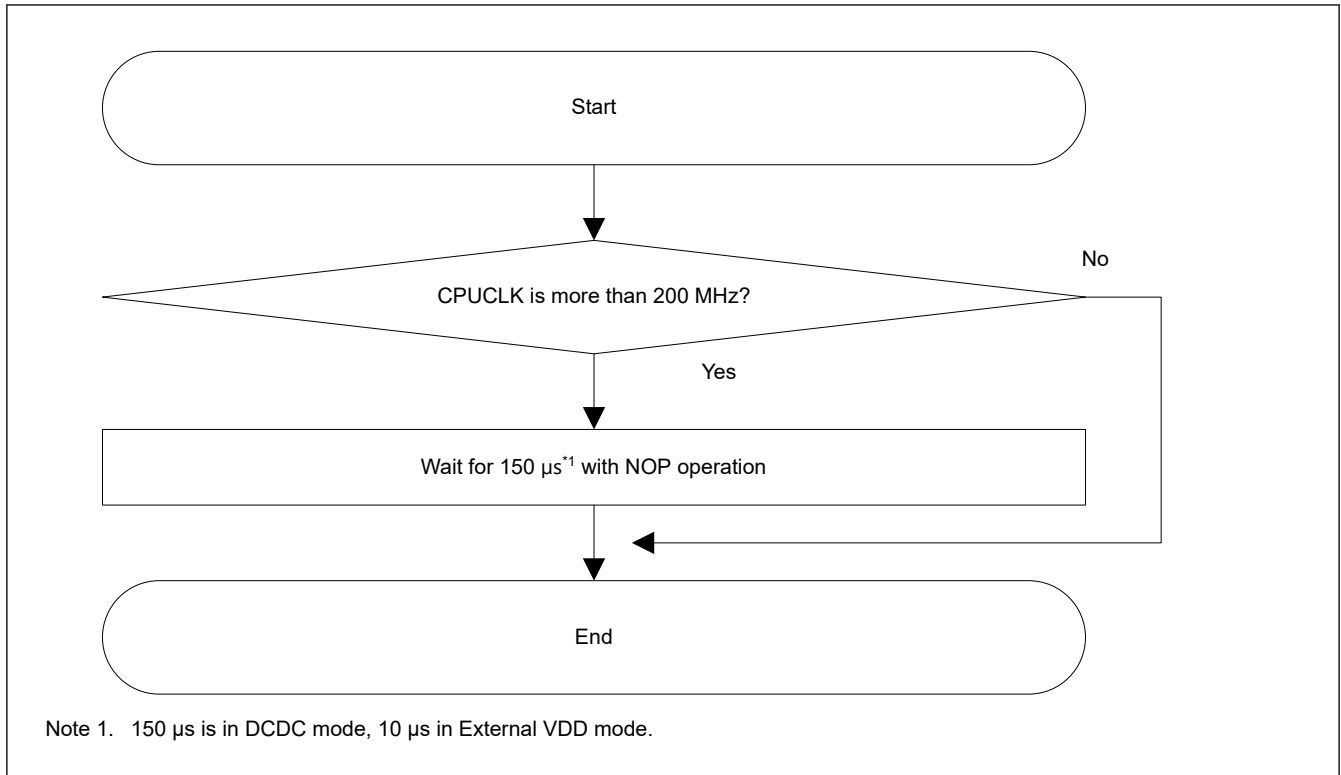


Figure 10.7 Additional flow for the return from CPU Deep Sleep mode according to CPUCLK frequency

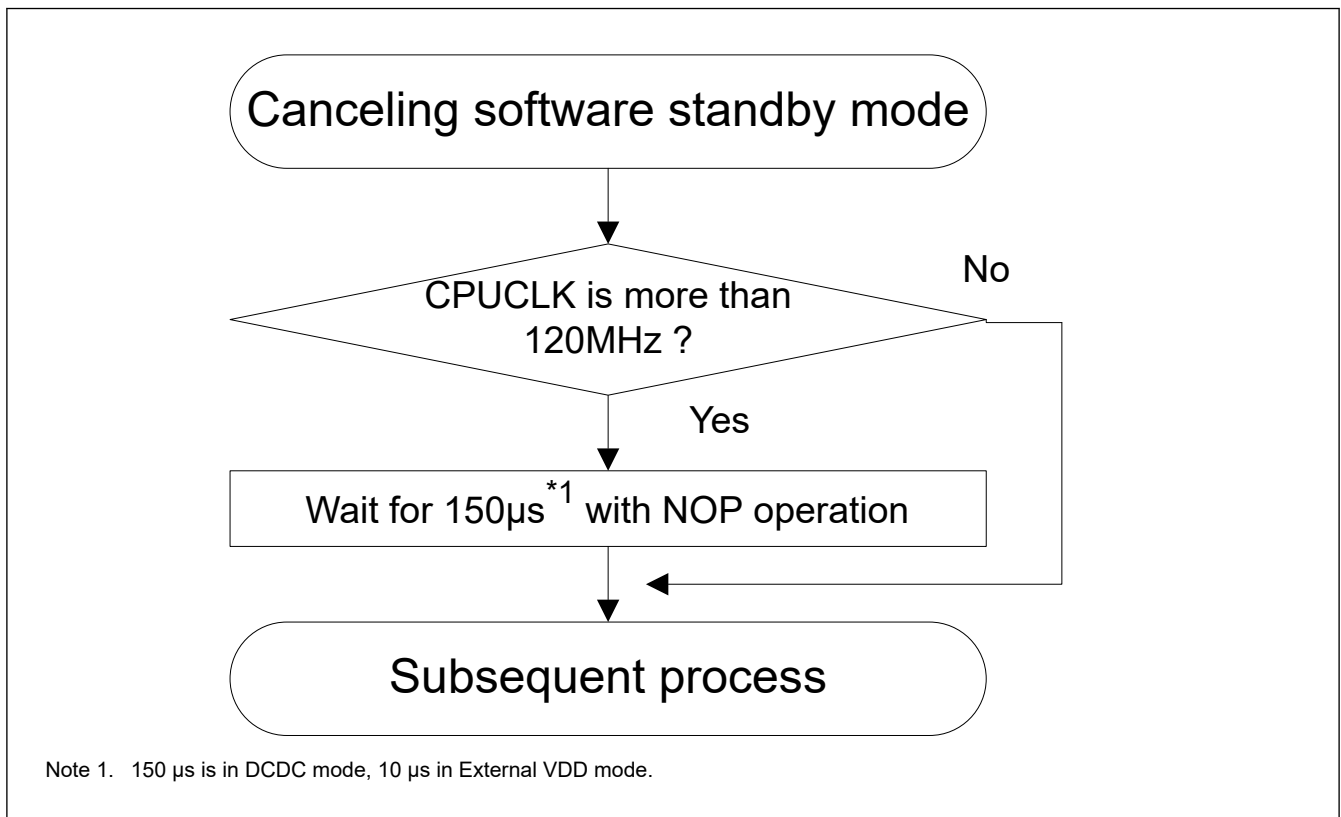


Figure 10.8 Additional flow for the canceling software standby mode

## 11. Battery Backup Function

### 11.1 Overview

The MCU provides a battery backup function that maintains partial battery powering in the event of a power loss. Switching between VCC and VBATT, the battery-powered area includes RTC, SOSC, backup register, tamper detection and VBATT\_R voltage drop detection. VBATT\_R is the output voltage of the battery power supply switch. This is the power supply of the battery powered area.

During normal operation, the battery-powered area is powered by the main power supply, the VCC pin. When a VCC voltage drop is detected, the power source switches to the dedicated battery backup power pin, the VBATT pin. When the voltage rises again, the power source switches back from VBATT to VCC.

#### 11.1.1 Battery backup function block diagram

Figure 11.1 shows the configuration of the battery backup function.

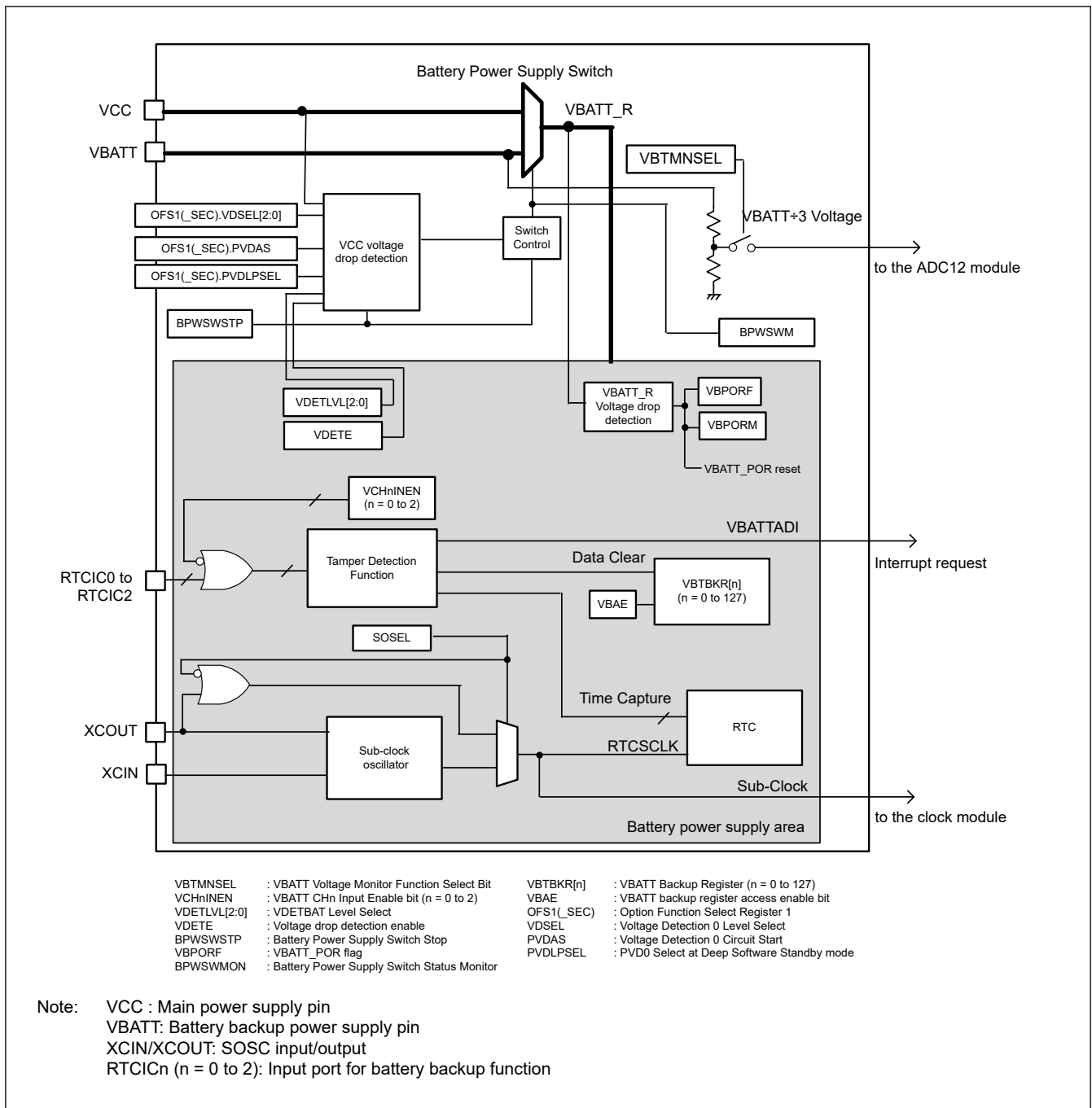


Figure 11.1 Battery Backup Function block diagram

### 11.1.2 Features of Battery Backup Function

The features include:

- Battery power supply switch
- VBATT\_R voltage drop detection function
- Backup registers
- Time capture pin detection
- Tamper Detection function
- VBATT voltage monitor function

### 11.1.3 Battery Power Supply Switch

When the voltage applied to the VCC pin drops, this feature switches the power supply from the VCC pin to the VBATT pin. When the voltage rises, it switches the power supply from the VBATT pin back to the VCC pin.

When VCC is lower than VDET<sub>BATT</sub>, and VCC is higher than VBAT, the injected current connects from the VCC to the VBATT pin through an internal diode. If the power supply battery connected to the VBATT pin cannot support this current injection, for example if the battery is not rechargeable, Renesas strongly recommends that you connect through a low-voltage threshold diode between the power supply battery and the VBATT pin.

It is necessary to enable voltage monitor 0 resets to use the battery backup function. The voltage monitor 0 level must be higher than the VBATT switch level.

### 11.1.4 VBATT\_R voltage drop detection function

VBATT\_R voltage drop detection function supports the battery-powered area. This function monitors the VBATT\_R voltage level. VBATT\_R is the output voltage of the battery power supply switch. This voltage drop detection causes a VBATT\_POR reset and initializes the battery-powered area. See details in each register description. The VBATT status register includes a flag to check for this voltage drop detections.

### 11.1.5 Backup Registers

The battery-powered area provides 128-byte backup registers. These registers retain data when VBATT is supplied and VCC is powered off. When tampering is detected, data of backup register can be cleared to 0x00.

### 11.1.6 Time Capture Function

The RTC detects RTCIC<sub>n</sub> (n = 0 to 2) pin input level change. For more information, see [section 24, Realtime Clock \(RTC\)](#).

### 11.1.7 Tamper Detection Function

The tamper detection function detects the RTCIC<sub>n</sub> (n = 0 to 2) pin input event. The input event is defined as a change of RTCIC<sub>n</sub> (n = 0 to 2) pin input level. The tamper detection flag is set to 1 by the input event. When interrupt is enabled and flag is set to 1, tamper detection interrupt is generated. When backup registers clear is enabled and flag is set to 1, the data of backup registers is cleared. Time Capture Function can select this flag as the source of the time capture trigger.

### 11.1.8 VBATT Voltage Monitor Function

VBATT voltage monitor function is to monitor the input voltage level to the VBATT pin. The voltage level can be monitored as analog signal. For more information, see [section 42, 12-Bit A/D Converter \(ADC12\)](#).

## 11.2 Register Descriptions

### 11.2.1 BBFSAR : Battery Backup Function Security Attribute Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0x3D0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	NONS EC4	NONS EC3	NONS EC2	NONS EC1	NONS EC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NONSEC0	Non Secure Attribute bit 0 Target register: VBATTMNSLR 0: Secure 1: Non Secure	R/W
1	NONSEC1	Non Secure Attribute bit 1 Target register: VBTBER 0: Secure 1: Non Secure	R/W
2	NONSEC2	Non Secure Attribute bit 2 Target register: VBTICTLR, VBTICTLR2, VBTIMONR 0: Secure 1: Non Secure	R/W
3	NONSEC3	Non Secure Attribute bit 3 Target register: VBTBPCR1, VBTBPCR2, VBTBPSR 0: Secure 1: Non Secure	R/W
4	NONSEC4	Non Secure Attribute bit 4 Target register: VBTADSR, VBTADCR1, VBTADCR2 0: Secure 1: Non Secure	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

Note: This register is write-protected by PRCR register.

The BBFSAR register controls the secure attribute of the battery backup function registers.

**NONSEC0 bit (Non Secure Attribute bit 0)**

This bit controls the security attribute of VBATTMNSLR.

**NONSEC1 bit (Non Secure Attribute bit 1)**

This bit controls the security attribute of VBTBER.

**NONSEC2 bit (Non Secure Attribute bit 2)**

This bit controls the security attribute of VBTICTLR, VBTICTLR2, and VBTIMONR.

**NONSEC3 bit (Non Secure Attribute bit 3)**

This bit controls the security attribute of VBTBPCR1 and VBTADCR2.

**NONSEC4 bit (Non Secure Attribute bit 4)**

This bit controls the security attribute of VBTADSR, VBTADCR1 and VBTADCR2.

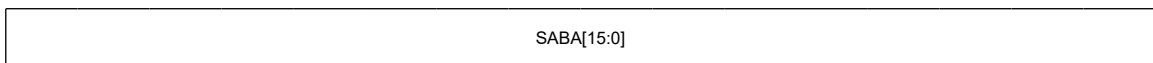
**11.2.2 VBR SABAR : VBATT Backup Register Security Attribute Boundary Address Register**

Base address: SYSC = 0x4001\_E000

Offset address: 0x3B0

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:



Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0

Bit	Symbol	Function	R/W
15:0	SABA[15:0]	Boundary address between secure and non-secure	R/W

Note: S-TYPE-6, P-TYPE-2

VBRSABAR specify the boundary address between Secure and Non-secure regions of VBATT Backup Register. This register specifies lower 16 bits of VBATT Backup Register address. Secure region is less than SABA. non-secure region is SABA or higher. The boundary address can set in units of 32 byte, so the value written from b4 to b0 should be 0. SABA has no effect other than VBATT backup register.

VBATT Backup register is separated as follows.

Secure region :  $0x4001\_ED00 \leq \text{Address} < 0x4001\_0000 + \text{SABA}$

Non-secure region :  $0x4001\_0000 + \text{SABA} \leq \text{Address}$

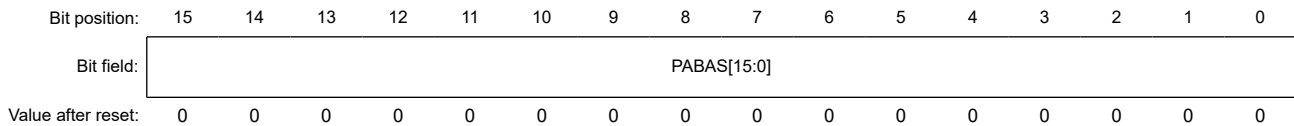
The initial value specifies an address greater than the end address of VBATT Backup Register, so all area is Secure.

If you specify all area of the VBATT Backup Register as non-secure, specify the top address in the VBATT Backup Register area.

### 11.2.3 VBRPABARS : VBATT Backup Register Privilege Attribute Boundary Address Register for Secure Region

Base address: SYSC = 0x4001\_E000

Offset address: 0x3B4



Bit	Symbol	Function	R/W
15:0	PABAS[15:0]	Boundary address between privileged and unprivileged.	R/W

Note: S-TYPE-6, P-TYPE-2

VBRPABARS specify the boundary address between Privileged and Unprivileged areas in Secure region of VBATT Backup Register. This register further separates the Secure region of VBATT Backup Register set by VBRSABAR register into Privileged and Unprivileged areas. This register specifies lower 16 bits of VBATT Backup Register address. Privileged area is less than PABAS. Unprivileged area is PABAS or higher. The boundary address can set in units of 32byte, so the value written from b4 to b0 should be 0. PABAS has no effect other than VBATT backup register.

VBATT Backup register is separated as follows.

Privileged area in Secure region :  $0x4001\_ED00 \leq \text{Address} < 0x4001\_0000 + \text{PABAS}$   
 $0x4001\_ED00 \leq \text{Address} < 0x4001\_0000 + \text{SABA}$

This area that satisfies both address conditions.

Unprivileged area in Secure region :  $0x4001\_0000 + \text{PABAS} \leq \text{Address} < 0x4001\_0000 + \text{SABA}$

The initial value specifies before the top address of the VBATT Backup Register, so all the Secure region is Unprivileged. In this case, the setting of this register is actually ignored.

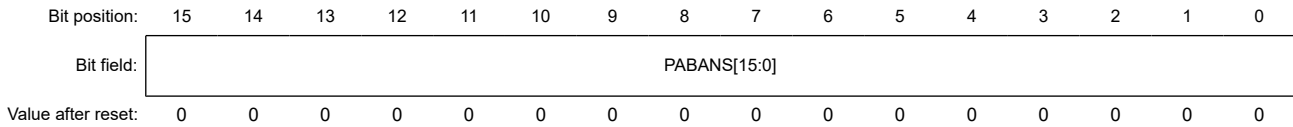
If you specify Privileged for the entire VBATT Backup Register that you have set as the Secure region, specify VBRPABARS register to the 0xFFE0 .

This is to narrow the verification space of the target specification base verification.

### 11.2.4 VBRPABARNS : VBATT Backup Register Privilege Attribute Boundary Address Register for Non-secure Region

Base address: SYSC\_NS = 0x5001\_E000

Offset address: 0x3B8



Bit	Symbol	Function	R/W
15:0	PABANS[15:0]	Boundary address between privileged and unprivileged.	R/W

Note: S-TYPE-7, P-TYPE-2

VBRPABARNS specify the boundary address between Privileged area and Unprivileged area in Non-secure region of VBATT Backup Register. This register further separates the Non-secure region of VBATT Backup Register set by VBRPABAR into Privileged and Unprivileged areas. This register specifies lower 16 bits of VBATT Backup Register address. Privileged area is less than PABANS. Unprivileged area is PABANS or higher. The boundary address can set in units of 32byte, so the value written from b4 to b0 should be 0. PABANS has no effect other than VBATT backup register.

VBATT Backup register is separated as follows.

Privileged area in Secure region :  $0x5001\_0000 + SABA \leq \text{Address} < 0x5001\_0000 + PABANS$

Unprivileged area in Secure region :  $0x5001\_0000 + PABANS \leq \text{Address}$   
 $0x5001\_0000 + SABA \leq \text{Address}$

This area that satisfies both address conditions

The initial value specifies before the top address of VBATT Backup Register, so all the Non-secure region is Unprivileged. In this case, the setting of this register is actually ignored.

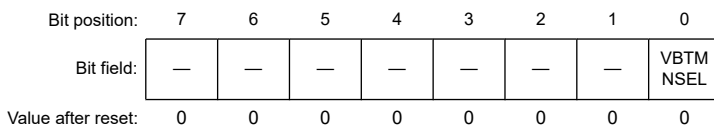
If you specify Privileged for the entire VBATT Backup Register that you have set as the Non-secure region, specify VBRPABARNS register to the 0xFFE0 .

This is to narrow the verification space of the target specification base verification

### 11.2.5 VBATTMNSCLR : Battery Backup Voltage Monitor Function Select Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0xA84



Bit	Symbol	Function	R/W
0	VBTMNSEL	VBATT Voltage Monitor Function Select Bit 0: Disables VBATT voltage monitor function 1: Enables VBATT voltage monitor function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

VBATTMNSCLR is the register which controls VBATT voltage monitor function.

This register is initialized by all reset sources except VBATT\_POR reset.



**VBTMNSEL bit (VBATT Voltage Monitor Function Select Bit)**

Select VBATT low voltage monitor function. After setting this bit to 1, it is necessary to wait  $t_{MONWT}$  for the monitor level to stabilize.

Consumption current increases while  $VBTMNSEL = 1$ . So, after monitoring the VBATT voltage level, clear  $VBTMNSEL$  to 0 in order to reduce power consumption of VBATT power supply.

For more information on  $t_{MONWT}$ , see [section 52, Electrical Characteristics](#)

**11.2.6 VBTBER : VBATT Backup Enable Register**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xC40

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	VBAE	—	—	—

Value after reset: 0 0 0 0 1 0 0 0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	VBAE	VBATT backup register access enable bit 0: Disable to access VBTBKR[n] 1: Enable to access VBTBKR[n]	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

VBAE bit is initialized by the power on reset of VCC or after transition to VBATT mode.

**VBAE bit (VBATT backup register access enable bit)**

You must write 1 to VBAE before accessing  $VBTBKR[n]$  ( $n = 0$  to 127) and you must write 0 to VBAE after finishing all access (write or read) to  $VBTBKR[n]$ . If you do not write 0 to VBAE, the data of  $VBTBKR[n]$  is not kept in VBATT mode.

Also, while VBAE is set to 0, the value of  $VBTBKR[n]$  can be retained even if the VDD power supply and VCC power supply are powered off.

To access  $VBTBKR[n]$ , wait for at least 500 ns after writing 1 to VBAE, and then access  $VBTBKR[n]$ .

Before entering the Deep Software Standby mode, you must write 0 to VBAE.

Wait at least 250 ns after writing 0 to VBAE, before entering the Deep Software Standby mode.

If it is not used  $VBTBKR[n]$ , It should be changed VBAE to 0 to reduce power consumption of  $VBTBKR[n]$ .

**11.2.7 VBTBKR[n] : VBATT Backup Register (n = 0 to 127)**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address:  $0xD00 + 0x001 \times n$

Bit position:	7	6	5	4	3	2	1	0
Bit field:								

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	n/a	VBATT Backup Register The value of this register is retained even in VBATT mode. This register is not initialized by any reset sources.	R/W

Note: S-TYPE-3, P-TYPE-3

### VBTBKR[n] bits (VBATT Backup Register)

The value of this register is retained even in VBATT mode.

You can use 32, 16 or 8-bit access instruction when accessing the VBTBKR[n] register. When accessing, please note that the byte order of the data stored in the VBTBKR[n] register is little endian.

The data of VBTBKR[n] register is cleared to 0x00 by VBATT\_POR reset or tamper detection function.

**Table 11.1 Address of VBATT Backup Register for Secure Region**

Address	Symbol
0x4001_ED00 to 0x4001_ED0F	VBTBKR[0] to VBTBKR[15]
0x4001_ED10 to 0x4001_ED1F	VBTBKR[16] to VBTBKR[31]
0x4001_ED20 to 0x4001_ED2F	VBTBKR[32] to VBTBKR[47]
0x4001_ED30 to 0x4001_ED3F	VBTBKR[48] to VBTBKR[63]
0x4001_ED40 to 0x4001_ED4F	VBTBKR[64] to VBTBKR[79]
0x4001_ED50 to 0x4001_ED5F	VBTBKR[80] to VBTBKR[95]
0x4001_ED60 to 0x4001_ED6F	VBTBKR[96] to VBTBKR[111]
0x4001_ED70 to 0x4001_ED7F	VBTBKR[112] to VBTBKR[127]

### 11.2.8 VBTICTLR : VBATT Input Control Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xC4C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VCH2INEN	VCH1INEN	VCH0INEN

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	VCH0INEN	VBATT CH0 Input Enable 0: RTCIC0 input disable 1: RTCIC0 input enable	R/W
1	VCH1INEN	VBATT CH1 Input Enable 0: RTCIC1 input disable 1: RTCIC1 input enable	R/W
2	VCH2INEN	VBATT CH2 Input Enable 0: RTCIC2 input disable 1: RTCIC2 input enable	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: Note: S-TYPE-3, P-TYPE-2

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

VBTICTLR is the register that can select RTCICn (n = 0 to 2) pin as input. This register is only initialized by VBATT\_POR reset.

### VCHnINEN bits (VBATT CHn Input Enable Bits) (n = 0 to 2)

The VCHnINEN bits enable the input direction on the RTCICn. A wait time of 50 μs is required for the operation to stabilize after the RTCICn pin input is enabled. If VCC voltage of the MCU is lower than 1.8V and these bits are set to 1, RTC registers and VBTBKR[n] (n = 0 to 127) can be accessed 20 μs after the setting of this bit.

### 11.2.9 VBTICTLR2 : VBATT Input Control Register 2

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xC4D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	VCH2 EG	VCH1 EG	VCH0 EG	—	VCH2 NCE	VCH1 NCE	VCH0 NCE

Value after reset: 0 1 1 1 0 0 0 0

Bit	Symbol	Function	R/W
0	VCH0NCE	VBATT CH0 Input Noise Canceler Enable 0: RTCIC0 pin input noise canceler disable 1: RTCIC0 pin input noise canceler enable	R/W
1	VCH1NCE	VBATT CH1 Input Noise Canceler Enable 0: RTCIC1 pin input noise canceler disable 1: RTCIC1 pin input noise canceler enable	R/W
2	VCH2NCE	VBATT CH2 Input Noise Canceler Enable 0: RTCIC2 pin input noise canceler disable 1: RTCIC2 pin input noise canceler enable	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	VCH0EG	VBATT CH0 Input Edge Select 0: RTCIC0 pin input event is detected on falling edge 1: RTCIC0 pin input event is detected on rising edge	R/W
5	VCH1EG	VBATT CH1 Input Edge Select 0: RTCIC1 pin input event is detected on falling edge 1: RTCIC1 pin input event is detected on rising edge	R/W
6	VCH2EG	VBATT CH2 Input Edge Select 0: RTCIC2 pin input event is detected on falling edge 1: RTCIC2 pin input event is detected on rising edge	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

VBTICTLR2 is the register that can select RTCICn (n = 0 to 2) pin input mode. This register is only initialized by VBATT\_POR reset.

#### VCHnNCE bits (VBATT CHn Input Noise Canceler Enable) (n = 0 to 2)

The VCHnNCE bit enables the input noise canceler on the RTCICn (n = 0 to 2) pin input.

#### VCHnEG bits (VBATT CHn Input Edge Select) (n = 0 to 2)

The VCHnEG bit selects input event detection edge to set VBTADF<sub>n</sub> flag.

### 11.2.10 VBTIMONR : VBATT Input Monitor Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xC4E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VCH2 MON	VCH1 MON	VCH0 MON

Value after reset: 0 0 0 0 0 0 x x x

Bit	Symbol	Function	R/W
0	VCH0MON	VBATT CH0 Input monitor 0: RTCIC0 pin input is low level 1: RTCIC0 pin input is high level.	R
1	VCH1MON	VBATT CH1 Input monitor 0: RTCIC1 pin input is low level 1: RTCIC1 pin input is high level.	R
2	VCH2MON	VBATT CH2 Input monitor 0: RTCIC2 pin input is low level 1: RTCIC2 pin input is high level.	R
7:3	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-2

VBTIMONR is the register that can monitor RTCICn (n = 0 to 2) pin input level.

### VCHnMON bits (VBATT CHn Monitor) (n = 0 to 2)

The VCHnMON bit indicates input level on the RTCICn (n = 0 to 2) pin.

## 11.2.11 VBTBPCR1 : VBATT Battery Power Supply Control Register 1

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA88

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	—	—	BPWS WSTP
------------	---	---	---	---	---	---	---	--------------

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	BPWSWSTP	Battery Power Supply Switch Stop 0: Battery power supply switch enable 1: Battery power supply switch stop	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTBPCR1 register controls battery power supply switch. This register is initialized by all reset sources except Deep Software Standby reset and VBATT\_POR reset.

### BPWSWSTP bit (Battery Power Supply Switch Stop)

The BPWSWSTP bit can enable switching the backup module supply power source from VCC to VBATT when the voltage applied to the VCC pin drops.

When stop is selected, the battery backup module power supply is always from VCC. Set the VDETE bit to 0 after setting the BPWSWSTP bit to 1. The BPWSWSTP bit must not be set from 1 to 0 while the VDETE bit is 1. If this setting is made, the state of the backup power area cannot be guaranteed.

## 11.2.12 VBTBPCR2 : VBATT Battery Power Supply Control Register 2

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xC45

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	—	VDET E	—	VDETLVL[2:0]	
------------	---	---	---	-----------	---	--------------	--

Value after reset: 0 0 0 0 0 1 1 0

Bit	Symbol	Function	R/W
2:0	VDETLVL[2:0]	V <sub>DETBAT</sub> Level Select 0 0 0: 2.80 V 0 0 1: 2.53 V 0 1 0: 2.10 V 0 1 1: 1.95 V 1 0 0: 1.85 V 1 0 1: 1.75 V 1 1 0: setting prohibited 1 1 1: setting prohibited	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	VDETE	Voltage drop detection enable 0: VCC Voltage drop detection disable 1: VCC Voltage drop detection enable	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTBPCR2 register controls battery power supply switch. This register is only initialized by VBATT\_POR reset.

#### VDETLVL[2:0] bits (V<sub>DETBAT</sub> Level Select)

The VDETLVL[2:0] bit selects V<sub>DETBAT</sub> level. When the voltage of the VCC pin drops below V<sub>DETBAT</sub>, the power supply source switches from the VCC pin to the VBATT pin. The V<sub>DETBAT</sub> level should be below Voltage monitoring 0 level.

This MCU can reduce power consumption of the Deep Software Standby mode 1 or 2. When the MCU enters the Deep Software Standby mode 1 or 2 and the low power consumption function of voltage monitor 0 is enabled by OFS1(\_SEC).PVDLPSEL bit and OFS1(\_SEC).PVDAS bit, the battery power supply switch is controlled by the voltage monitor 0.

The VDETLVL[2:0] bit should change while the VDETE bit is 0. The VCC voltage detection function needs t<sub>DETWT</sub> wait time for stabilization at the change of V<sub>DETBAT</sub> level. The VDETE bit should be set to 1 after this wait time.

For details on VDETLVL[2:0] bit, see [section 11.3.2. VBATT Battery Power Supply Switch Usage](#).

For more information on t<sub>DETWT</sub>, see [section 52, Electrical Characteristics](#).

#### VDETE bit (Voltage drop detection enable)

The VDETE bit enables the VCC voltage drop detection function for the battery power supply switch. The initial value of the VDETE bit is 0, at this time, the VCC voltage drop detection function for the battery power supply switch is disabled. If you use the battery power supply switch function, you must select V<sub>DETBAT</sub> level and enables the VCC voltage drop detection function.

For details on VDETE bit, see [section 11.3.2. VBATT Battery Power Supply Switch Usage](#).

### 11.2.13 VBTBPSR : VBATT Battery Power Supply Status Register

Base address: SYSC = 0x4001\_E000  
 SYSC\_NS = 0x5001\_E000

Offset address: 0xC46

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	BPWS WM	VBPO RM	—	—	—	VBPO RF
Value after reset:	0	0	x	x	0	0	0	x

Bit	Symbol	Function	R/W
0	VBPORF	VBATT_POR Flag 0: VBATT_R voltage drop is not detected 1: VBATT_R voltage drop is detected	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	VBPORM	VBATT_POR Monitor 0: VBATT_R voltage < $V_{PDR(BATR)}$ 1: VBATT_R voltage > $V_{PDR(BATR)}$	R
5	BPWSWM	Battery Power Supply Switch Status Monitor 0: VCC voltage < $V_{DET\text{BATT}_m}$ 1: VCC voltage > $V_{DET\text{BATT}_m}$	R
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTBPSR register indicate battery power supply status.

### VBPORF bit (VBATT\_POR Flag)

The VBPORF flag indicates that VBATT\_R voltage drop was detected and VBATT\_POR reset was asserted.

[Setting condition]

- When VBATT\_R voltage drops below  $V_{PDR(BATR)}$ .

[Clearing conditions]

- When 0 is written.

### VBPORM bit (VBATT\_POR Monitor)

The VBPORM bit indicates comparison result between VBATT\_R and  $V_{PDR(BATR)}$ .

### BPWSWM bit (Battery Power Supply Switch Status Monitor)

The BPWSWM bit indicates comparison result between VCC and  $V_{DET\text{BATT}_m}$  ( $m = 0$  to 6).

## 11.2.14 VBTADSR : VBATT Tamper detection Status Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xC48

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VBTA DF2	VBTA DF1	VBTA DF0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VBTADF0	VBATT Tamper Detection flag 0 0: RTCIC0 input edge is not detected 1: RTCIC0 input edge is detected	R/W
1	VBTADF1	VBATT Tamper Detection flag 1 0: RTCIC1 input edge is not detected 1: RTCIC1 input edge is detected	R/W
2	VBTADF2	VBATT Tamper Detection flag 2 0: RTCIC2 input edge is not detected 1: RTCIC2 input edge is detected	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTADSR register indicate tamper detection function status. This register is only initialized by VBATT\_POR reset.

### VBTADF<sub>n</sub> flags (VBATT Tamper Detection flag n) (n = 0, 1, 2)

The VBTADF<sub>n</sub> ( $n = 0, 1, 2$ ) flag indicates that RTCIC<sub>n</sub> ( $n = 0, 1, 2$ ) input edge is detected. The edge type of RTCIC<sub>n</sub> ( $n = 0, 1, 2$ ) input can be selected by VCH<sub>n</sub>EG ( $n = 0, 1, 2$ ) bit.

[Setting condition]

- When RTCICn (n = 0, 1, 2) input edge is detected.

[Clearing conditions]

- When 0 is written after 1 is read from VBTADFn (n = 0, 1, 2) flag.

### 11.2.15 VBTADCR1 : VBATT Tamper detection Control Register 1

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xC49

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	VBTA DCE2	VBTA DCE1	VBTA DCE0	—	VBTA DIE2	VBTA DIE1	VBTA DIE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VBTADIE0	VBATT Tamper Detection Interrupt Enable 0 0: Interrupt by VBTADF0 flag is disable 1: Interrupt by VBTADF0 flag is enable	R/W
1	VBTADIE1	VBATT Tamper Detection Interrupt Enable 1 0: Interrupt by VBTADF1 flag is disable 1: Interrupt by VBTADF1 flag is enable	R/W
2	VBTADIE2	VBATT Tamper Detection Interrupt Enable 2 0: Interrupt by VBTADF2 flag is disable 1: Interrupt by VBTADF2 flag is enable	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	VBTADCE0	VBATT Tamper Detection Backup Register Clear Enable 0 0: Clear Backup Register by VBTADF0 flag is disable 1: Clear Backup Register by VBTADF0 flag is enable	R/W
5	VBTADCE1	VBATT Tamper Detection Backup Register Clear Enable 1 0: Clear Backup Register by VBTADF1 flag is disable 1: Clear Backup Register by VBTADF1 flag is enable	R/W
6	VBTADCE2	VBATT Tamper Detection Backup Register Clear Enable 2 0: Clear Backup Register by VBTADF2 flag is disable 1: Clear Backup Register by VBTADF2 flag is enable	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTADCR1 register control tamper detection functions. This register is only initialized by VBATT\_POR reset.

#### **VBTA DIEn bits (VBATT Tamper Detection Interrupt Enable) (n = 0, 1, 2)**

The VBTADIEn (n = 0, 1, 2) bit enables Tamper detection interrupt.

#### **VBTA DCEn bits (VBATT Tamper Detection Backup Register Clear Enable) (n = 0, 1, 2)**

The VBTADCEn (n = 0, 1, 2) bit enables backup register clearing by tamper detection flag.

## 11.2.16 VBTADCR2 : VBATT Tamper detection Control Register 2

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xC4A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	VBRT CES2	VBRT CES1	VBRT CES0

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	VBRTCES0	VBATT RTC Time Capture Event Source Select 0 0: RTCIC0 1: VBTADF0	R/W
1	VBRTCES1	VBATT RTC Time Capture Event Source Select 1 0: RTCIC1 1: VBTADF1	R/W
2	VBRTCES2	VBATT RTC Time Capture Event Source Select 2 0: RTCIC2 1: VBTADF2	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

The VBTADCR2 register control tamper detection functions. This register is only initialized by VBATT\_POR reset.

### VBRTCESn bits (VBATT RTC Time Capture Event Source Select) (n = 0, 1, 2)

The VBRTCESn (n = 0, 1, 2) bit selects RTC time capture event source.

## 11.3 Operation

### 11.3.1 Battery Backup Function

When the voltage on the VCC pin drops, power can be supplied to the VBATT\_R backup power area from the VBATT pin. The power supply from the VCC pin is resumed when the voltage on the VCC pin exceeds  $V_{DET\_BATT\_m}$  (m = 0 to 6). When a drop of power supply from VCC pin is detected, backup power area enter the VBATT Mode. In the VBATT Mode, power supply is switched to VBATT pin. The power supply from the VCC pin is resumed when the voltage on the VCC pin exceeds  $V_{DET\_BATT\_m}$  (m = 0 to 6). This power supply change does not affect the VBATT\_R backup power area function.

It is necessary to enable voltage monitor 0 reset to use the battery backup

The VBATT\_R backup power area include following functions:

- RTC (including time capture detection, triggered by a change of the time capture pin input level)
- Sub-clock oscillator (including XCIN and XCOOUT pins)
- VBATT Backup Register
- Tamper Detection Function
- VBATT Voltage Monitor Function

Table 11.2 shows the operating states in VBATT mode.

**Table 11.2 Operating States in VBATT Mode (1 of 2)**

Operating state	VBATT Mode
Transition condition	Detection of VCC voltage drop
Canceling method other than reset	Detection of VCC voltage rise
State after cancellation by an interrupt	—



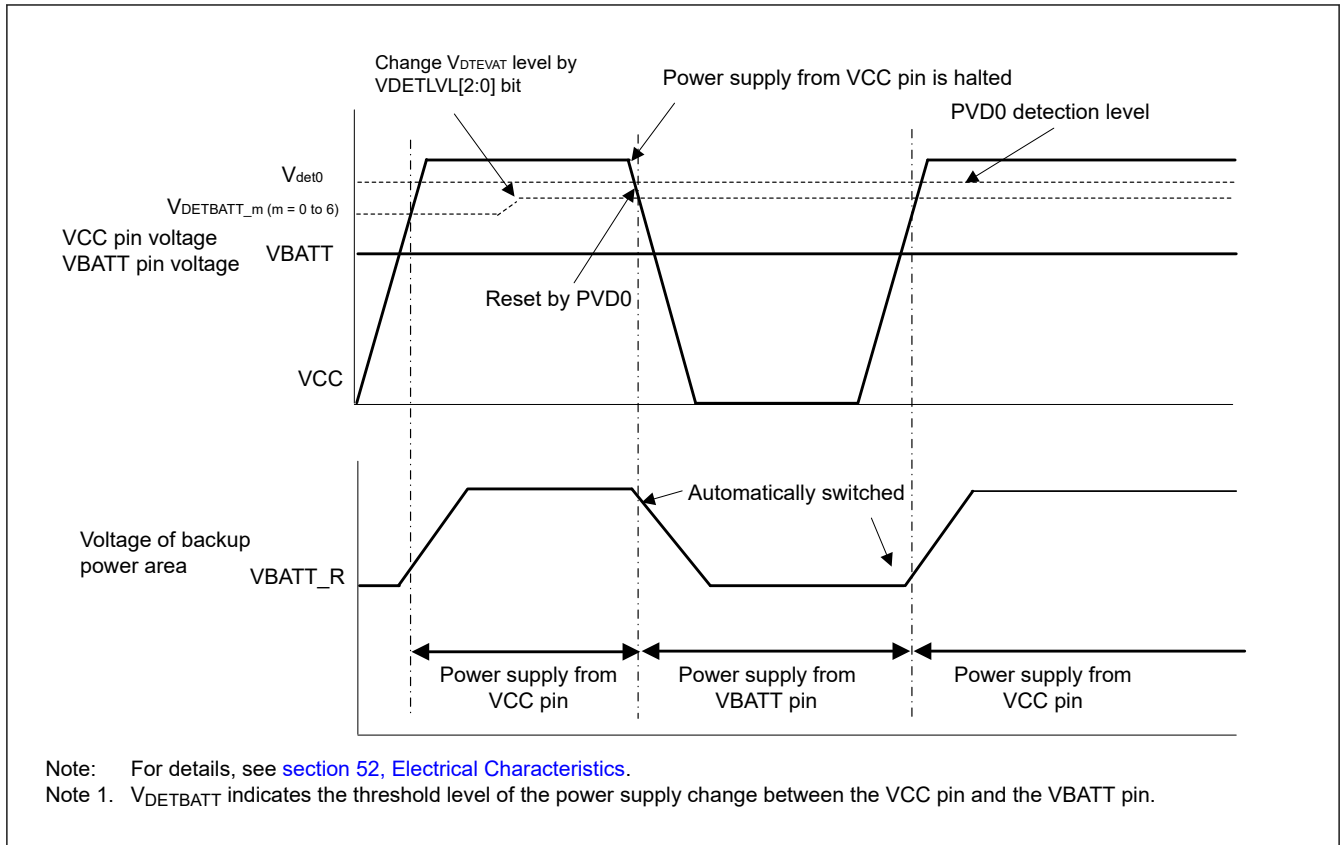
**Table 11.2 Operating States in VBATT Mode (2 of 2)**

Operating state	VBATT Mode
State after cancellation by a reset	—
Main clock oscillator	Stop
Sub-clock oscillator	Operating or not operating can be selected by SOSCCR.SOSTP bit. The status of the oscillator is same as before entering VBATT mode.
High-speed on-chip oscillator	Stop
Middle-speed on-chip oscillator	Stop
Low-speed on-chip oscillator	Stop
PLL1	Stop
PLL2	Stop
CPU	Stop (Undefined)
SRAM	Stop (Undefined)
Standby SRAM	Stop (Undefined)
VBATT Backup Register	Stop (Retained or zeroized is selectable when Tamper input was detected.)
Flash memory	Stop (Retained)
Realtime clock (RTC)	Selectable when selecting clock which is operating as the count source.
Programmable voltage detection circuit (PVD)	Stop
Power-on reset circuit	Stop
Other Peripheral modules	Stop (Undefined)
I/O ports	RTCICn ports (n = 0 to 2): Selectable EXCIN: Selectable All ports not specified here: Undefined

Note: Stop (Retained) means that the contents of the internal registers are retained but the operations are suspended.

Note: Stop (Undefined) means that the contents of the internal registers are undefined and power to the internal circuit is cut off.

Figure 11.2 shows switching sequence of Battery backup function.



**Figure 11.2 Switching sequence of battery backup function**

### 11.3.2 VBATT Battery Power Supply Switch Usage

The battery power supply switch can switch the power supply from the VCC pin to the VBATT pin when the voltage being applied to the VCC pin drops. When the voltage rises, this switch changes the power supply from the VBATT pin to the VCC pin.

The VCC voltage drop detection function for the battery power supply switch in "Figure 11.1" consists of two voltage drop detectors. One is a VCC drop detector for normal operation of the battery power supply switch that operates according to the settings of the VDETTLVL[2:0] bit and the VDETE bit. The other is voltage monitoring 0 to reduce power consumption during the Deep Software Standby mode 1 or 2.

In cold start, VDETE bit is initialized by VBATT\_POR reset and the VCC voltage drop detection function for the battery power supply switch is disabled.

To use the battery power supply switch, the proper  $V_{DET\text{BATT}_m}$  ( $m = 0$  to 6) level must be selected by VDETTLVL[2:0] bits and wait  $t_{DETWT}$  for stabilization. After wait, the VCC voltage drop detection function is enabled by VDETE bit. The battery power supply switch has a constraint that the  $V_{DET\text{BATT}_m}$  ( $m = 0$  to 6) level selected by VDETTLVL[2:0] bits should be below Voltage monitoring 0 level.

When the MCU transitions to the Deep Software Standby mode 1 or 2 and the low power consumption function of voltage monitor 0 is enabled, the VCC drop detector for normal operation of the battery power supply switch is stopped to reduce power consumption. At this time, the power supply switch is controlled by the voltage monitor 0. This control by voltage monitor 0 continues until it returns from Deep Software Standby mode.

The battery backup function should be used after the voltage monitoring 0 reset is enabled (OFS1(\_SEC).PVDAS bit is 0).

When the low power consumption function of voltage monitor 0 is enabled by OFS1(\_SEC).PVDLPSEL bit, the voltage monitor 1 and 2 must be disabled before enter the Deep Software Standby mode 1 or 2.

If you don't use battery power supply switch. You must set BPWSWSTP bit to 1 and short the VCC and VBATT pins. When BPWSWSTP bit is 1, switch is stop, and power is always supplied from VCC pin.

### 11.3.3 VBATT\_R voltage drop detection function Usage

This function enables to detect VBATT\_R voltage drop. When VBATT\_R voltage is drop and lower than  $V_{PDR(BATR)}$ , VBATT\_POR reset is asserted. The VBPFORF flag is set to 1 by VBATT\_POR reset. If VBPFORF flag is 1, RTC counter and register is invalid and Battery Backup Function registers are reset. You must initialize the functions included backup power area.

Figure 11.3 shows VBATT\_R voltage drop detection function.

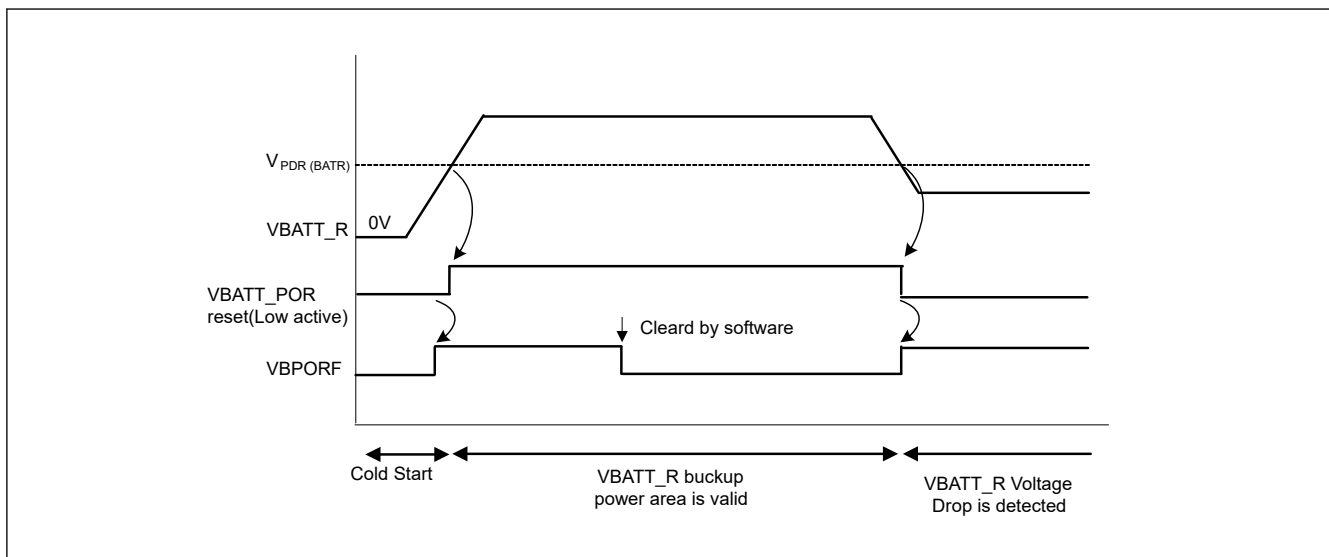


Figure 11.3 VBATT\_R voltage drop detection function

### 11.3.4 VBATT Backup Register Usage

You can use 32, 16 or 8-bit access instruction when accessing the VBATT backup registers.

However, for example, when a 32bit access instruction is executed, 8-bit read or write operation is executed with 4 consecutive times. When accessing, please note that the byte order of the data stored in the VBTBKR[n] ( $n = 0$  to 127) register is little endian.

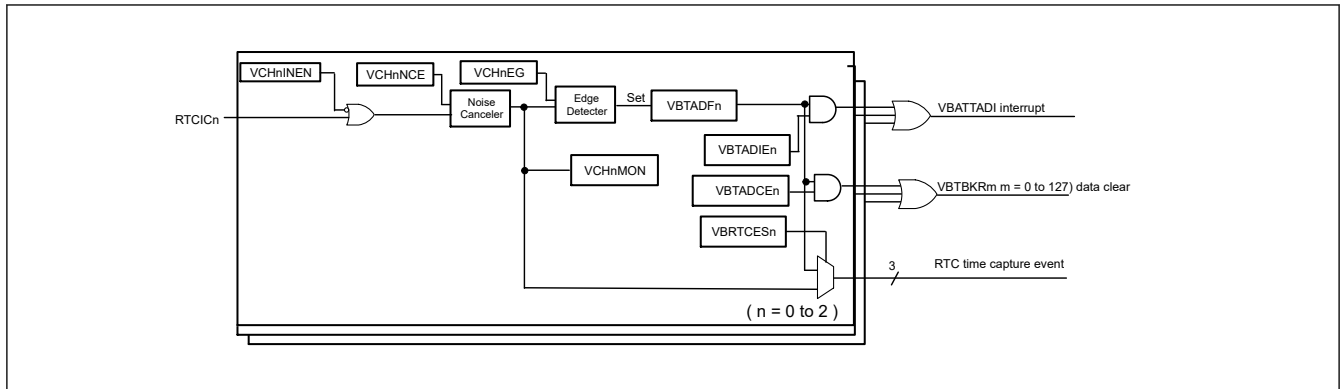
The data of VBATT backup register is cleared to 0x00 by VBATT\_POR reset or tamper detection function.

The tamper detection function starts clearing operation of the VBATT backup register when the clear function is enabled and the tamper detection flag is set to 1. Do not cancel the clearing operation for 100ns after starting. To cancel the clearing operation, disable the clear function or clear the tamper detection flag to 0. After canceling the clearing operation, do not access the VBATT backup register for 500ns.

### 11.3.5 Tamper Detection Usage

The tamper detection function detects the RTCICn ( $n = 0$  to 2) pin input event. The input event is defined as a change of RTCICn pin input level. The VBTADF $n$  ( $n = 0$  to 2) flag is set to 1 by the input event. When generate interrupt is enabled and the flag is set to 1, tamper detection interrupt is generated. When backup register clear function is enabled and flag is set to 1, the data of VBATT backup register is cleared. RTC time capture event source can select RTCICn pin input or the VBTADF $n$  flag. If you need to synchronize time capture with a set of VBTADF $n$  flags to 1, select the VBTADF $n$  flag as the RTC time capture event source is recommended.

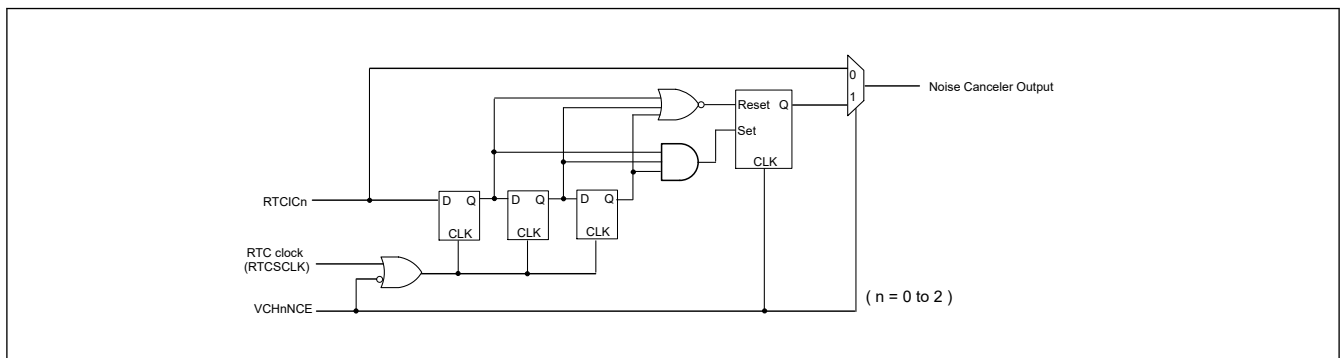
Figure 11.4 shows Tamper detection function.



**Figure 11.4 Tamper detection function**

RTCICn pin inputs is enabled by the VCHnINEN (n = 0 to 2) bit. A wait time of 50us is required for the operation to stabilize after the RTCICn input is enabled.

The VCHnNCE (n = 0 to 2) bit enables noise canceler. The noise canceler samples input signals at the Sub-Clock (RTCSCLK) and removes the pulses whose length is less than three sampling cycles. Figure 11.5 shows noise canceler circuit block diagram.



**Figure 11.5 Tamper detection function**

After enabling this noise canceler, a stabilization time of 5 clocks is required.

The VCHnEG (n = 0 to 2) bit enables to select Tamper detection edge. The edge detector operate asynchronously. When these control registers are changed, The VBTADFn flag may be set to 1 in a pseudo. The VBTADFn flag need check and clear to 0 after initialization of control registers . The VCHNnMON (n = 0 to 2) bit enables to monitor current input status. This bit also need check for inactive level after initialization of control registers .

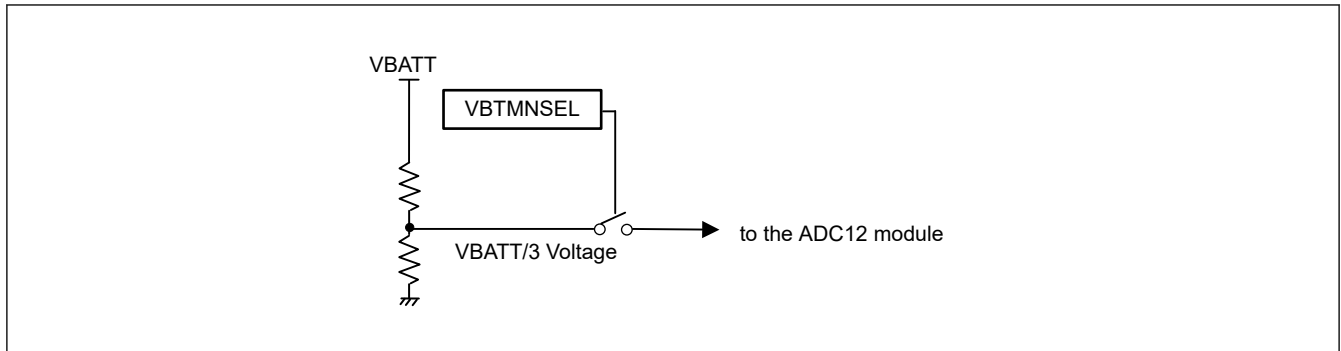
The VBATTADI interrupt is asserted when 1 or more of 3 channels are flagged and interrupt is enabled.

The VBTBKRM(m = 0 to 127) register data is cleared to 0x00 when 1 or more of 3 channels are flagged and the VBATT backup register clear function is enabled.

### 11.3.6 VBATT voltage monitor function Usage

You can monitor the input voltage level of VBATT pin. 1/3 of VBATT pin voltage level can be monitored as analog signal. After setting the VBTMNSEL bit to 1, it is necessary to wait t<sub>MONWT</sub> for the monitor level to stabilize. For details on t<sub>MONWT</sub>, see section 52, Electrical Characteristics.

Figure 11.6 shows VBATT voltage monitor function.



**Figure 11.6 VBATT voltage monitor function**

Note: For more information, see [section 42, 12-Bit A/D Converter \(ADC12\)](#).

Note: When VBTMNSEL bit is 1, VBAT power consumption is increased. It is recommended that VBTMNSEL bit sets to 1 only monitor timing.

### 11.3.7 Initial Settings Examples

#### 11.3.7.1 Cold start and using the power supply switch flow example

When turning on VCC pin and VBATT pin power for the first time, use the following flow to initialize.

1. Check VBPORM bit. If VBPORM flag is 0, wait until it changes to 1.
2. Clear VBPORF flag to 0.
3. Set the VDETLVL[2:0] bit to appropriate value. The  $V_{DET\text{BATT}_m}$  level is Selected.
4. Wait  $t_{DETWT}$  for the VCC voltage detection function stabilization.
5. Set the VDETE bit to 1. The VCC voltage detection function is enabled.
6. Enable Sub-Clock Oscillator if needed.
7. Set Other Battery Backup Function and RTC registers.

For details on  $t_{DETWT}$ , see [section 52, Electrical Characteristics](#).

For details on Sub-Clock Oscillator, see [section 8, Clock Generation Circuit](#).

#### 11.3.7.2 Warm start and using the power supply switch flow example

The MCU transitions to VBATT mode when the power of the VCC pin is turned off while the power of the VBATT pin is on. In this section, warm start means when the power supply of the VCC pin is turned on in VBATT mode. When the MCU starts processing with a warm start, execute the following flow in order to check the VBATT\_R voltage drop before executing other process.

1. Check VBPORM flag. If VBPORM flag is 0, wait until it changes to 1.
2. Check VBPORF flag. If VBPORF flag is 1, the VBATT\_R voltage drop is detected. Branch "[section 11.3.7.1. Cold start and using the power supply switch flow example 2](#)" to re-initialize the backup power area. If not, it means that the VBATT\_R voltage does not drop and the state of the backup power area is retained. So, you do not need to re-initialize the backup power area.

#### 11.3.7.3 Not using the power supply switch flow example

When not using the power supply switch, VCC pin and VBATT pin should be shorted. In this case, the VBATT\_POR reset cannot follow the power on reset of VCC and VBATT\_R voltage drop may not be detected. So, initialize the backup power supply area with the following flow.

1. Set the BPWSWSTP bit to 1. The power supply switch is stopped.
2. Check VBPORM flag. If VBPORM flag is 1, wait until it changes to 0.
3. Clear the VDETE bit to 0. The VCC voltage drop detection function is stopped.

4. Clear the VDETLVL[2:0] bit to 110b. The initial value is selected.
5. Check VBPORF flag. When VBPORF flag is 1, clear it to 0.
6. Set the SOSTP bit to 1 regardless of its value. Stop Sub-Clock Oscillator.
7. Initialize the VBTICTLR register and SOMCR.SOSEL bit. Initialization is recommended because these registers are related to the control of the IO port.
8. Initialize Other Battery Backup Function registers if need. Other Battery Backup Function registers are VBTICTLR2, VBTADSR, VBTADCR1, VBTADCR2, and VBTBKR[n].
9. Enable Sub-Clock Oscillator if needed.
10. Set RTC registers.

For details on Sub-Clock Oscillator, see [section 8, Clock Generation Circuit](#).

#### 11.3.7.4 Tamper detection function initialization setting flow example

If using the tamper detection function, it is recommended to initialize according to the following flow example.

1. Set the VCHnINEN to appropriate value.
2. Wait 50us to stabilize after the RTCICn input.
3. Set the VCHnNCE bit and the VCHnEG bit.
4. Wait 5 RTC-clocks to stabilize , if noise canceler is enabled.
5. Checking the VCHnMON bit indicates inactive is recommended. If it is active, it may not be possible to detect the tamper.
6. Initialize the invalid status of VBTADFn flag by clear to 0 after reading dummy read.
7. Set VBTADCR1, VBTADCR2 to enable interrupt, backup register clear, and RTC time capture event.
8. enables RTC time capture function if needed.

## 11.4 Interrupt Sources

The Battery Backup Function has two interrupt sources and are listed in [Table 11.3](#).

**Table 11.3 Battery Backup Function interrupt Sources**

Symbol	Interrupt source	Interrupt flag	Interrupt conditions
VBATTADI	VBATT Tamper detection	VBTADF0	VBTADF0 = 1, VBTADIE0 = 1
		VBTADF1	VBTADF1 = 1, VBTADIE1 = 1
		VBTADF2	VBTADF2 = 1, VBTADIE2 = 1

## 11.5 Usage Notes

1. Operation of the sub-clock oscillator and RTC are not guaranteed when the voltage level on VBATT is lower than the guaranteed operation range. Initialize the RTC when the VBATT pin falls below the guaranteed operating voltage and then powers up again.
2. A reset generated while writing to registers described in this section might destroy the register value.
3. When VCC is higher than  $V_{DET\_BATT\_m}$ , the VCC pin and VBATT pin are separated. When VCC is lower than  $V_{DET\_BATT}$  and the switch is connected to the VBATT pin, and if the voltage on VBATT drops lower than VCC, current might flow into the VBATT pin through the parasitic diode between the VCC and VBATT pins.
4. During RTC operation using the voltage from the VBATT pin and the I/O ports within the backup, the power supply area can only be used as time capture event input pins for the RTC.

## 12. Register Write Protection

### 12.1 Overview

The register write protection function protects important registers from being overwritten due to software errors. The registers to be protected are set with the Protect Register (PRCR\_S and PRCR\_NS).

The two protected registers work on one Secure set registers/bits and the other for Non-secure set registers/bits. They are collectively mentioned as PRCR.

Table 12.1 lists the association between the bits in the PRCR register and the registers to be protected.

The register information to which the PRCR bit is applied is provided in the description of each register.

**Table 12.1 Association between the bits in the PRCR register and registers to be protected**

PRCR bit	Register to be protected
PRC0	<ul style="list-style-type: none"> <li>Registers related to the clock generation circuit: SCKDIVCR, SCKDIVCR2, SCKSCR, PLLCCR, PLLCR, MOSCCR, HOCOCCR, HOCOCCR2, MOCOCCR, FLLCR1, FLLCR2, CKOCR, OSTDCR, OSTDSR, PLL2CCR, PLL2CR, PLLCCR2, PLL2CCR2, SCICKDIVCR, SCICKCR, SPICKDIVCR, SPICKCR, MOCOUTCR, HOCOUTCR, USBCKDIVCR, CANFDCKDIVCR, USBCKCR, CANFDCKCR, MOSCSCR, HOCOSCR, MOSCWTCR, MOMCR, SOSCCR, SOMCR, LOCOCR, LOCOUTCR, SYRACCR</li> </ul>
PRC1	<ul style="list-style-type: none"> <li>Registers related to the low power modes: SBYCR, OPCCR, PDRAMSCR0, PDRAMSCR1, SSCR1, LPSCR, DPSBYCR, DPSWCR, DPSIER0-3, DPSIFR0-3, DPSIEGR0-2, PLL1LDOCR, PLL2LDOCR, HOCOLDOCR, LVOCR</li> <li>Register related to the battery backup function: VBTBER, VBTICTLR, VBTBKR[n] (n = 0 to 127), VBTBPCR1, VBTBPCR2, VBTBPSR, VBTADSR, VBTADCR1, VBTADCR2, VBTICTLR2</li> </ul>
PRC3	<ul style="list-style-type: none"> <li>Registers related the PVD: PVD1CR1, PVD1SR, PVD2CR1, PVD2SR, PVD1CMPCR, PVD2CMPCR, PVD1CR0, PVD2CR0, PVD1FCR, PVD2FCR, VBATTMNSLR</li> </ul>
PRC4	<ul style="list-style-type: none"> <li>Registers related to the Security and Privilege setting registers: ELCSARx (x=A,B)<sup>†1</sup>, ELCPARx (x=A,B), PSARx (x=B to E), MSSAR, PPARx (x=B to E), MSPAR, PmSAR (m=0 to 9), CPUSAR, DEBUGSAR, ICUSARx (x=A,B,E to I), SRAMSAR, BUSSARx(x=A,B), MMPUSARx (x=A,B), DTCSAR, DMACSAR, DMACCHSAR, DMACCHPAR, TEVTRCR, SRAMSABAR1, STBRAMSABAR, STBRAMPABAR_NS, STBRAMPABAR_S, FSAR, CGFSAR, RSTSAR, LPMSAR, PVDSAR, BBFSAR, DPFSAR, RSCSAR, VBRASABAR, VBRPABARS, VBRPABARNS</li> </ul>
PRC5 <sup>†1</sup>	<ul style="list-style-type: none"> <li>Registers related to the reset control SYRSTMSK0, SYRSTMSK2</li> </ul>

Note 1. Only PRCR\_S is supported.

### 12.2 Register Descriptions

#### 12.2.1 PRCR\_S : Protect Register for Secure (PRCR\_S)

Base address: SYSC = 0x4001\_E000

Offset address: 0x3FA

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]							—	—	PRC5	PRC4	PRC3	—	PRC1	PRC0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRC0	Enable writing to the registers related to the clock generation circuit 0: Disable writes 1: Enable writes	R/W

Bit	Symbol	Function	R/W
1	PRC1	Enable writing to the registers related to the low power modes, and the battery backup function 0: Disable writes 1: Enable writes	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	PRC3	Enable writing to the registers related to the PVD 0: Disable writes 1: Enable writes	R/W
4	PRC4	Enables writing to the registers related to the security and privilege setting registers. 0: Disable writes 1: Enable writes	R/W
5	PRC5	Enables writing to the registers related to the reset control. 0: Disable writes 1: Enable writes	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	0xA5: Enables writing to the PRCR_S register. Other than the above: Disables writing to the PRCR_S register. 0x00 are readable if these bits are read.	W

Note: S-TYPE6, P-TYPE2

PRCR\_S is used to protect registers that are always Secure or configured as Secure.

### PRCn bits (Protect bit n) (n = 0, 1, 3, 4, 5)

The PRCn bits enable or disable writing to the protected registers listed in [Table 12.1](#). Setting the PRCn bits to 1 enables writing, and to 0 disables writing.

The register controlled by PRC4 may not reflect the PRC4 change when PRCR\_S and its controlled registers are continuously written access. Avoid continuous write access or read the PRCR\_S after PRC4 change, and then write to the register controlled by PRC4.

### PRKEY[7:0] (Register write protection key bit)

These bits control permission and prohibition of writing to the PRCR\_S register. To write PRCn bits of PRCR\_S register, write 0xA5 to the PRKEY[7:0]. In case of writing other than 0xA5 to PRKEY[7:0], PRCn bits do not be changed even if writing to the PRCR\_S register.

## 12.2.2 PRCR\_NS : Protect Register for Non-secure (PRCR\_NS)

Base address: SYSC\_NS = 0x5001\_E000

Offset address: 0x3FE

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]							—	—	—	PRC4	PRC3	—	PRC1	PRC0	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRC0	Enable writing to the registers related to the clock generation circuit 0: Disable writes 1: Enable writes	R/W
1	PRC1	Enable writing to the registers related to the low power modes, and the battery backup function 0: Disable writes 1: Enable writes	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W



Bit	Symbol	Function	R/W
3	PRC3	Enable writing to the registers related to the PVD 0: Disable writes 1: Enable writes	R/W
4	PRC4	Enables writing to the registers related to the security and pprivilege setting registers. 0: Disable writes 1: Enable writes	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	0xA5: Enables writing to the PRCR_NS register. Other than the above: Disables writing to the PRCR_NS register. 0x00 are readable if these bits are read.	W

Note: S-TYPE7, P-TYPE2

PRCR\_NS is used to protect registers that are configured as Non-secure.

### PRCn bits (Protect bit n) (n = 0, 1, 3, 4)

The PRCn bits enable or disable writing to the protected registers listed in [Table 12.1](#). Setting the PRCn bits to 1 enables writing, and to 0 disables writing.

The register controlled by PRC4 may not reflect the PRC4 change when PRCR\_NS and its controlled registers are continuously written access. Avoid continuous write to that registers or read the PRCR\_NS after PRC4 change, and then write to the register controlled by PRC4.

### PRKEY (Register write protection key bit)

These bits control permission and prohibition of writing to the PRCR\_NS register. To write PRCn bits of PRCR\_NS register, write 0xA5 to the PRKEY[7:0]. In case of writing other than 0xA5 to PRKEY[7:0], PRCn bits do not be changed even if writing to the PRCR\_NS register.

## 13. Interrupt Controller Unit (ICU)

### 13.1 Overview

The Interrupt Controller Unit (ICU) controls which event signals are linked to the Nested Vector Interrupt Controller (NVIC), the DMA Controller (DMAC), and the Data Transfer Controller (DTC) modules. The ICU also controls non-maskable interrupts.

[Table 13.1](#) lists the ICU specifications, [Figure 13.1](#) shows a block diagram, and [Table 13.2](#) lists the I/O pins.

**Table 13.1 ICU specifications**

Parameter		Description
Maskable interrupts	Peripheral function interrupts	<ul style="list-style-type: none"> <li>Interrupts from peripheral modules</li> <li>Number of sources: 234 (select factor within event list numbers 17 to 511)</li> </ul>
	External pin interrupts	<ul style="list-style-type: none"> <li>Interrupt detection on low level<sup>*4</sup>, falling edge, rising edge, rising and falling edges. One of these detection methods can be set for each source</li> <li>Digital filter function supported</li> <li>16 sources, with interrupts from IRQi (i = 0 to 15) pins.</li> </ul>
	Interrupt requests to CPU (NVIC)	<ul style="list-style-type: none"> <li>96 interrupt requests are output to NVIC.<sup>*5</sup></li> </ul>
	DMAC control	<ul style="list-style-type: none"> <li>The DMAC can be activated using interrupt sources<sup>*1</sup></li> <li>The target interrupt source can be selected individually for every DMAC channels.</li> </ul>
	DTC control	<ul style="list-style-type: none"> <li>The DTC can be activated using interrupt sources<sup>*1</sup></li> <li>The method for selecting an interrupt source is the same as that of the interrupt request to NVIC.</li> </ul>
Non-maskable interrupts <sup>*2</sup>	NMI pin interrupt	<ul style="list-style-type: none"> <li>Interrupt from the NMI pin</li> <li>Interrupt detection on falling edge or rising edge</li> <li>Digital filter function supported</li> </ul>
	Oscillation stop detection interrupt <sup>*3</sup>	Interrupt on detecting that the main oscillation has stopped
	WDT underflow/refresh error <sup>*3</sup>	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	IWDT underflow/refresh error <sup>*3</sup>	Interrupt on an underflow of the down-counter or occurrence of a refresh error
	Voltage-monitoring 1 interrupt <sup>*3</sup>	Voltage monitor 1 interrupt of the voltage monitor 1 circuit (PVD_PVD1)
	Voltage-monitoring 2 interrupt <sup>*3</sup>	Voltage monitor 2 interrupt of the voltage monitor 2 circuit (PVD_PVD2)
	Common memory error interrupt	Common memory errors include SRAM parity error, or Standby SRAM parity error
	Bus error Interrupt	Bus error includes MPU and TZF error
	CPU Lockup error interrupt	CPU Lockup error
Security	Secure	Some registers have Security Attribution
	Privilege	Each register of the ICU can only be accessed with Privilege access
Low power modes	<ul style="list-style-type: none"> <li>CPU Sleep mode: return is initiated by non-maskable interrupts or any other interrupt source</li> <li>CPU Deep Sleep and Software Standby mode: Return is initiated by non-maskable interrupts. Interrupt can be selected as WUPEN register.</li> </ul> <p>See <a href="#">section 13.2.15. WUPEN0 : Wake Up Interrupt Enable Register 0</a>, <a href="#">section 13.2.16. WUPEN1 : Wake Up interrupt enable register 1</a>.</p>	
TrustZone Filter	Available	

Note 1. For the DMAC and DTC activation sources, see [section 13.3.2. Event Number](#).

Note 2. Non-maskable interrupts can be enabled only once after a reset release.

Note 3. These non-maskable interrupts can also be used as maskable interrupts. When used as maskable interrupts, do not change the value of the NMIER register from the reset state. To enable voltage monitor 1 and voltage monitor 2 interrupts, set the PVD1CR1.IRQSEL and PVD2CR1.IRQSEL bits to 1.

Note 4. Low level: interrupt detection is not canceled if you do not clear it after a detection.

Note 5. Appropriate alignment of the vector table according to the number of interrupts to be used is required. For example, in case of using maximum number of interrupts available in this products, at least 128-words alignment must be set.

Figure 13.1 shows the ICU block diagram.

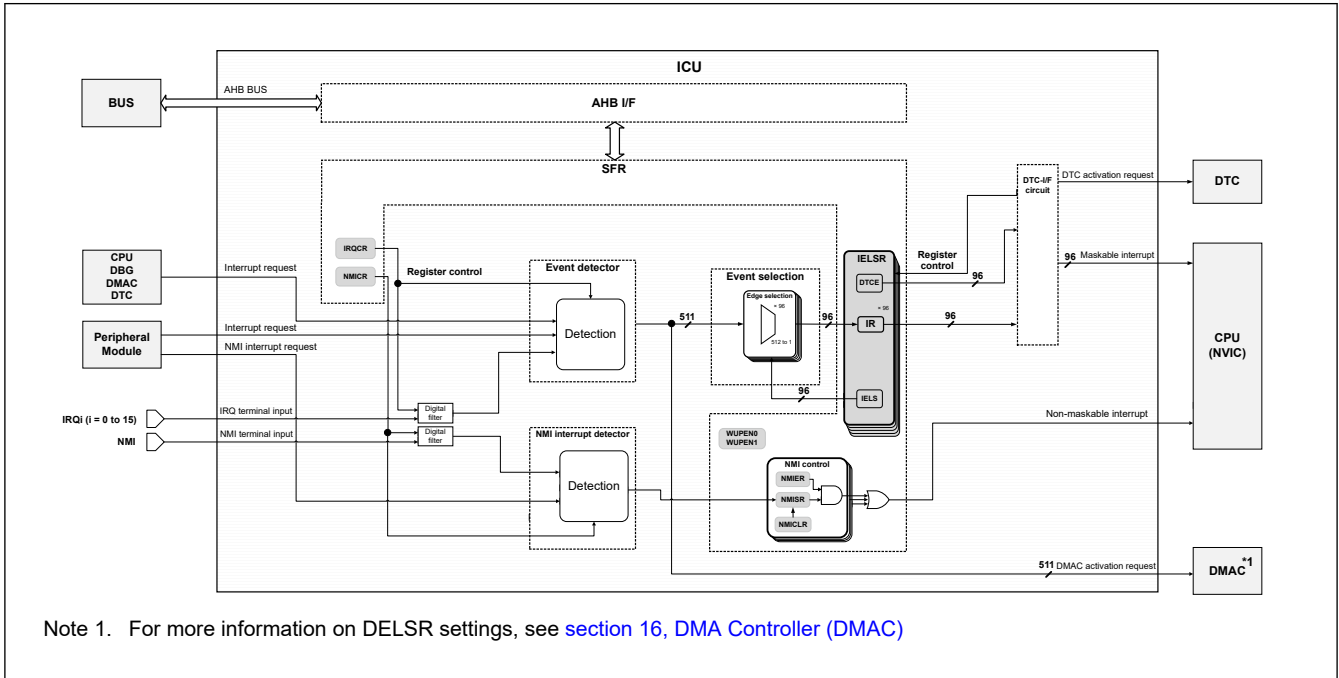


Figure 13.1 ICU block diagram

Table 13.2 lists the ICU input/output pins.

Table 13.2 ICU I/O pins

Pin name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQi (i = 0 to 15)	Input	External interrupt request pins

### 13.2 Register Descriptions

This chapter does not describe the Arm® NVIC internal registers. For information about these registers, see Arm Limited., Arm® Cortex®-M85 Processor Technical Reference Manual (101924\_0002\_05\_en).

#### 13.2.1 ICUSARA : Interrupt Controller Unit Security Attribution Register A

Base address: CPSCU = 0x4000\_8000  
 CPSCU\_NS = 0x5000\_8000

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIRQ CR15	SAIRQ CR14	SAIRQ CR13	SAIRQ CR12	SAIRQ CR11	SAIRQ CR10	SAIRQ CR9	SAIRQ CR8	SAIRQ CR7	SAIRQ CR6	SAIRQ CR5	SAIRQ CR4	SAIRQ CR3	SAIRQ CR2	SAIRQ CR1	SAIRQ CR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SAIRQCR15 to SAIRQCR0	Security attributes of registers for the IRQCRn register 0: Secure 1: Non-secure	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE1, P-TYPE1

Note: This register is write-protected by PRCR\_S.PRC4 register

### SAIRQCRn bits (Security attributes of registers for the IRQCRn register)

The target registers are as follows:

- IRQCR0 to IRQCR15 registers
- WUPEN0.IRQWUPEN[15:0] bits

## 13.2.2 ICUSARB : Interrupt Controller Unit Security Attribution Register B

Base address: CPSCU = 0x4000\_8000  
CPSCU\_NS = 0x5000\_8000

Offset address: 0x44

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

Bit	Symbol	Function	R/W
0	SANMI	Security attributes of registers for nonmaskable interrupt 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE1, P-TYPE1

Note: This register is write-protected by PRCR\_S.PRC4 register

### SANMI bit (Security attributes of registers for nonmaskable interrupt)

Security attributes of registers for non-maskable interrupt. The target registers are as follows:

- NMISR
- NMIER
- NMICLR
- NMICR

The value of AIRCR.BFHFNMINS bit [13] in Application Interrupt and Reset Control Register of Arm CPU should be the same as the value of security attribution. The initial values of AIRCR.BFHFNMINS and the SANMI bits are different. AIRCR.BFHFNMINS is secure and SANMI is non-secure. Polarity has the same meaning so program these to match.

Note: Only one of Secure and Non-secure can set security attribution for non-maskable interrupt-related registers. If you program the Secure attribute as secure, it always goes to the Secure interrupt handler. To release any of the Non-maskable interrupt sources to the non-secure user, write a function to execute a nonsecure program from the interrupt handler for Secure.

### 13.2.3 ICUSARE : Interrupt Controller Unit Security Attribution Register E

Base address: CPSCU = 0x4000\_8000  
 CPSCU\_NS = 0x5000\_8000

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIIC0 WUP	SAAG T1CB WUP	SAAG T1CA WUP	SAAG T1UD WUP	SAUS BFS0 WUP	—	SART CPRD WUP	SART CALM WUP	—	—	—	SAVB ATTW UP	SAPV D2WU P	SAPV D1WU P	—	SAIW DTWU P
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	SAIWDTWUP	Security attributes of registers for WUPEN0.b16 0: Secure 1: Non-secure	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W
18	SAPVD1WUP	Security attributes of registers for WUPEN0.b18 0: Secure 1: Non-secure	R/W
19	SAPVD2WUP	Security attributes of registers for WUPEN0.b19 0: Secure 1: Non-secure	R/W
20	SAVBATTWUP	Security attributes of registers for WUPEN0.b20 0: Secure 1: Non-secure	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	SARTCALMWUP	Security attributes of registers for WUPEN0.b24 0: Secure 1: Non-secure	R/W
25	SARTCPRDWUP	Security attributes of registers for WUPEN0.b25 0: Secure 1: Non-secure	R/W
26	—	This bit is read as 0. The write value should be 0.	R/W
27	SAUSBFS0WUP	Security attributes of registers for WUPEN0.b27 0: Secure 1: Non-secure	R/W
28	SAAGT1UDWUP	Security attributes of registers for WUPEN0.b28 0: Secure 1: Non-secure	R/W
29	SAAGT1CAWUP	Security attributes of registers for WUPEN0.b29 0: Secure 1: Non-secure	R/W
30	SAAGT1CBWUP	Security attributes of registers for WUPEN0.b30 0: Secure 1: Non-secure	R/W
31	SAIIC0WUP	Security attributes of registers for WUPEN0.b31 0: Secure 1: Non-secure	R/W

Note: S-TYPE1, P-TYPE1

Note: This register is write-protected by PRCR\_S.PRC4 register

### 13.2.4 ICUSARF : Interrupt Controller Unit Security Attribution Register F

Base address: CPSCU = 0x4000\_8000  
 CPSCU\_NS = 0x5000\_8000

Offset address: 0x54

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SAUL P1BW UP	SAUL P1AW UP	SAUL P1UW UP	—	SAUL P0BW UP	SAUL P0AW UP	SAUL P0UW UP	—	—	—	—	SACO MPHS 0WUP	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	SACOMPMS0WUP	Security attributes of registers for WUPEN1.b3 0: Secure 1: Non-secure	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	SAULP0UWUP	Security attributes of registers for WUPEN1.b8 0: Secure 1: Non-secure	R/W
9	SAULP0AWUP	Security attributes of registers for WUPEN1.b9 0: Secure 1: Non-secure	R/W
10	SAULP0BWUP	Security attributes of registers for WUPEN1.b10 0: Secure 1: Non-secure	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	SAULP1UWUP	Security attributes of registers for WUPEN1.b12 0: Secure 1: Non-secure	R/W
13	SAULP1AWUP	Security attributes of registers for WUPEN1.b13 0: Secure 1: Non-secure	R/W
14	SAULP1BWUP	Security attributes of registers for WUPEN1.b14 0: Secure 1: Non-secure	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE1, P-TYPE1  
 Note: This register is write-protected by PRCR\_S.PRC4 register

### 13.2.5 ICUSARG : Interrupt Controller Unit Security Attribution Register G

Base address: CPSCU = 0x4000\_8000  
CPSCU\_NS = 0x5000\_8000

Offset address: 0x70

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIEL SR31	SAIEL SR30	SAIEL SR29	SAIEL SR28	SAIEL SR27	SAIEL SR26	SAIEL SR25	SAIEL SR24	SAIEL SR23	SAIEL SR22	SAIEL SR21	SAIEL SR20	SAIEL SR19	SAIEL SR18	SAIEL SR17	SAIEL SR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIEL SR15	SAIEL SR14	SAIEL SR13	SAIEL SR12	SAIEL SR11	SAIEL SR10	SAIEL SR9	SAIEL SR8	SAIEL SR7	SAIEL SR6	SAIEL SR5	SAIEL SR4	SAIEL SR3	SAIEL SR2	SAIEL SR1	SAIEL SR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIELSR31 to SAIELSR0	Security attributes of registers for IELSR31 to IELSR0 0: Secure 1: Non-secure	R/W

Note: S-TYPE1, P-TYPE1

Note: This register is write-protected by PRSCR\_S.PRC4 register

#### SAIELSRn bits (Security attributes of registers for IELSR31 to IELSR0)

The Secure Attribute managed within the Arm CPU NVIC must match the security attribution of IELSEn ( n = 0 to 31 ). NVIC internal registers are in NVIC\_ITNS0[31:0]. Polarity has the same meaning so program these to match.

### 13.2.6 ICUSARH : Interrupt Controller Unit Security Attribution Register H

Base address: CPSCU = 0x4000\_8000  
CPSCU\_NS = 0x5000\_8000

Offset address: 0x74

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIEL SR63	SAIEL SR62	SAIEL SR61	SAIEL SR60	SAIEL SR59	SAIEL SR58	SAIEL SR57	SAIEL SR56	SAIEL SR55	SAIEL SR54	SAIEL SR53	SAIEL SR52	SAIEL SR51	SAIEL SR50	SAIEL SR49	SAIEL SR48
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIEL SR47	SAIEL SR46	SAIEL SR45	SAIEL SR44	SAIEL SR43	SAIEL SR42	SAIEL SR41	SAIEL SR40	SAIEL SR39	SAIEL SR38	SAIEL SR37	SAIEL SR36	SAIEL SR35	SAIEL SR34	SAIEL SR33	SAIEL SR32
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIELSR63 to SAIELSR32	Security attributes of registers for IELSR63 to IELSR32 0: Secure 1: Non-secure	R/W

Note: S-TYPE1, P-TYPE1

Note: This register is write-protected by PRSCR\_S.PRC4 register

#### SAIELSRn bits (Security attributes of registers for IELSR63 to IELSR32)

The Secure Attribute managed within the Arm CPU NVIC must match the security attribution of IELSEn ( n = 32 to 63 ). NVIC internal registers are in NVIC\_ITNS1[31:0]. Polarity has the same meaning so program these to match.

### 13.2.7 ICUSARI : Interrupt Controller Unit Security Attribution Register I

Base address: CPSCU = 0x4000\_8000  
CPSCU\_NS = 0x5000\_8000

Offset address: 0x78

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SAIEL SR95	SAIEL SR94	SAIEL SR93	SAIEL SR92	SAIEL SR91	SAIEL SR90	SAIEL SR89	SAIEL SR88	SAIEL SR87	SAIEL SR86	SAIEL SR85	SAIEL SR84	SAIEL SR83	SAIEL SR82	SAIEL SR81	SAIEL SR80
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SAIEL SR79	SAIEL SR78	SAIEL SR77	SAIEL SR76	SAIEL SR75	SAIEL SR74	SAIEL SR73	SAIEL SR72	SAIEL SR71	SAIEL SR70	SAIEL SR69	SAIEL SR68	SAIEL SR67	SAIEL SR66	SAIEL SR65	SAIEL SR64
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SAIELSR95 to SAIELSR64	Security attributes of registers for IELSR95 to IELSR64 0: Secure 1: Non-secure	R/W

Note: S-TYPE1, P-TYPE1

Note: This register is write-protected by PRCR\_S.PRC4 register

#### SAIELSRn bits (Security attributes of registers for IELSR95 to IELSR64)

The Secure Attribute managed within the Arm CPU NVIC must match the security attribution of IELSEn ( n = 64 to 95 ). NVIC internal registers are in NVIC\_ITNS2[31:0]. Polarity has the same meaning so program these to match.

### 13.2.8 TEVTRCR : Trusted Event Route Control Register

Base address: CPSCU = 0x4000\_8000  
CPSCU\_NS = 0x5000\_8000

Offset address: 0x600

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

Bit	Symbol	Function	R/W
0	TEVTE	Trusted Event Route Control Register for IELSRn, DELSRn and ELCSRn 0: Disable 1: Enable	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE1, P-TYPE1

Note: This register is write-protected by PRCR\_S.PRC4

#### TEVTE bit (Trusted Event Route Control Register for IELSRn, DELSRn and ELCSRn)

When TEVTE = 1, the IELS [8:0] bits of IELSRn ( n = 0 to 95 ), all DELS [8:0] bits of DMAC.DELSRn ( n = 0 to 7 ), and all ELS [8:0] bits of ELC.ELSRn ( n = 0 to 18 ) are allowed secure access write. non-secure access write is protected. Additionally, when TEVTE = 1, if the Security Attribution of the target register ( IELS[31:16] bits of IELSRn ( n = 0 to 95 )) is non-secure, then secure access is not allowed. At this time, the upper level [31:16] cannot be read and write, but the response is OK, and no error occurs.



- IELSRn.IELS[8:0] (n = 0 to 95)
- DMAC.DELSRn.DELS[7:0] (n = 0 to 7)
- ELC.ELSRn.IELS[8:0] (n = 0 to 18)

### 13.2.9 IRQCRi : IRQ Control Register i (i = 0 to 15)

Base address: ICU\_COMMON = 0x4000\_6000  
ICU\_COMMON\_NS = 0x5000\_6000

Offset address: 0x000 + 0x1 × i

Bit position:	7	6	5	4	3	2	1	0
Bit field:	FLTEN	—	FCLKSEL[1:0]	—	—	—	IRQMD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	IRQMD[1:0]	IRQi Detection Sense Select 0 0: Falling edge 0 1: Rising edge 1 0: Rising and falling edges 1 1: Low level	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
5:4	FCLKSEL[1:0]	IRQi Digital Filter Sampling Clock Select 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	FLTEN	IRQi Digital Filter Enable 0: Digital filter is disabled 1: Digital filter is enabled.	R/W

Note: S-TYPE3, P-TYPE2

IRQCRi register changes must satisfy the following conditions:

- For a CPU interrupt or DTC trigger:  
Change the IRQCRi register value before setting the target IELSRn register (n = 0 to 95).  
The register value should be changed only when the value of the target IELSRn register is 0x0000.
- For a DMAC trigger:  
Change the IRQCRi register value before setting the target DMAC.DELSRn register (n = 0 to 7).  
The register value should be changed only when the value of the target DELSRn register is 0x0000.
- For a wakeup enable signal:  
Change the IRQCRi register setting before setting the target WUPEN0.IRQWUPEN[n] (n = 0 to 15). The register value should be changed when the target WUPEN0.IRQWUPEN[n] is 0.

#### IRQMD[1:0] bits (IRQi Detection Sense Select)

The IRQMD[1:0] bits set the detection sensing method for the IRQi external pin interrupt sources. For setting method when using external pin interrupt, see [section 13.5.7. External Pin Interrupts](#).

#### FCLKSEL[1:0] bits (IRQi Digital Filter Sampling Clock Select)

The FCLKSEL[1:0] bits select the digital filter sampling clock for the IRQi external pin interrupt request pins, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every 8 cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles)

For details of the digital filter, see [section 13.5.6. Digital Filter](#).

### FLTEN bit (IRQi Digital Filter Enable)

The FLTEN bit enables the digital filter used for the IRQi external pin interrupt sources. The digital filter is enabled when the IRQCRi.FLTEN bit is 1 and disabled when the IRQCRi.FLTEN bit is 0. The IRQi pin level is sampled at the clock cycle specified in the IRQCRi.FCLKSEL[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes. For details of the digital filter, see [section 13.5.6. Digital Filter](#).

## 13.2.10 NMISR : Non-Maskable Interrupt Status Register

Base address: ICU = 0x4000\_C000  
ICU\_NS = 0x5000\_C000

Offset address: 0x120

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LUST	—	CMST	BUSST	—	—	—	—	NMIST	OSTST	—	—	PVD2ST	PVD1ST	WDTST	IWDTST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTST	IWDT Underflow/Refresh Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
1	WDTST	WDT Underflow/Refresh Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
2	PVD1ST	Voltage Monitor 1 Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
3	PVD2ST	Voltage Monitor 2 Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
5:4	—	These bits are read as 0.	R
6	OSTST	Main Clock Oscillation Stop Detection Interrupt Status Flag 0: Interrupt not requested for main clock oscillation stop 1: Interrupt requested for main clock oscillation stop	R
7	NMIST	NMI Pin Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
11:8	—	These bits are read as 0.	R
12	BUSST	Bus Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
13	CMST	Common Memory Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R
14	—	This bit is read as 0.	R
15	LUST	CPU Lockup Error Interrupt Status Flag 0: Interrupt not requested 1: Interrupt requested	R

Note: S-TYPE3, P-TYPE2

The NMISR register monitors the status of non-maskable interrupt sources. Writes to the NMISR register are ignored. The setting in the Non-Maskable Interrupt Enable Register (NMIER) does not affect the status flags in this register. Before the end of the non-maskable interrupt handler, check that all of the bits in this register are set to 0 to confirm that no other NMI requests are generated during handler processing.

**IWDTST flag (IWDT Underflow/Refresh Error Interrupt Status Flag)**

The IWDTST flag indicates an IWDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.IWDTCLR bit.

[Setting condition]

When the IWDT underflow/refresh error interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.IWDTCLR bit.

**WDTST flag (WDT Underflow/Refresh Error Interrupt Status Flag)**

The WDTST flag indicates a WDT underflow/refresh error interrupt request. It is read-only and cleared by the NMICLR.WDTCLR bit.

[Setting condition]

When the WDT underflow/refresh error interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.WDTCLR bit.

**PVD1ST flag (Voltage Monitor 1 Interrupt Status Flag)**

The PVD1ST flag indicates a request for voltage monitor 1 interrupt. It is read-only and cleared by the NMICLR.PVD1CLR bit.

[Setting condition]

When the voltage monitor 1 interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.PVD1CLR bit.

**PVD2ST flag (Voltage Monitor 2 Interrupt Status Flag)**

The PVD2ST flag indicates a request for voltage monitor 2 interrupt. It is read-only and cleared by the NMICLR.PVD2CLR bit.

[Setting condition]

When the voltage monitor 2 interrupt is generated and this interrupt source is enabled.

[Clearing condition]

When 1 is written to the NMICLR.PVD2CLR bit.

**OSTST flag (Main Clock Oscillation Stop Detection Interrupt Status Flag)**

The OSTST flag indicates a main clock oscillation stop detection interrupt request. It is read-only and cleared by the NMICLR.OSTCLR bit.

[Setting condition]

When the main clock oscillation stop detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.OSTCLR bit.

**NMIST flag (NMI Pin Interrupt Status Flag)**

The NMIST flag indicates an NMI pin interrupt request. It is read-only and cleared by the NMICLR.NMISTCLR bit.

[Setting condition]

When an edge specified by the NMICR.NMIMD bit is input to the NMI pin.

[Clearing condition]

When 1 is written to the NMICLR.NMISTCLR bit.

**BUSST flag (Bus Error Interrupt Status Flag)**

The BUSST flag indicates a bus error interrupt request. It is read-only and cleared by the NMICLR.BUSCLR bit.

Bus error includes MPU and TZF errors.

[Setting condition]

When the bus error detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.SPECLR bit.

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status is set again even if this status is cleared, and an NMI request is issued to the CPU. If the CPU returns from the NMI handler, the CPU jumps back to the NMI handler.

In the case of level detection, use the following steps to clear the Status flag.

- (1) Negate the level of an input factor.
- (2) Perform a peripheral read access once and make sure the level interrupt is cleared.
- (3) Clear the status flag by NMICLR.BUSCLR.

**CMST flag (Common Memory Error Interrupt Status Flag)**

The CMST flag indicates common memory error interrupt request. It is read-only and cleared by the NMICLR.CMCLR bit.

Common memory errors include SRAM Parity or StandbyRAM Parity.

[Setting condition]

When the common memory error detection interrupt is generated.

[Clearing condition]

When 1 is written to the NMICLR.CMCLR bit.

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status is set again even if this status is cleared, and an NMI request is issued to the CPU. If the CPU returns from the NMI handler, the CPU jumps back to the NMI handler.

In the case of level detection, use the following steps to clear the Status flag.

- (1) Negate the level of an input factor.
- (2) Perform a peripheral read access once and make sure the level interrupt is cleared.
- (3) Clear the status flag by NMICLR.CMCLR.

**LUST flag (CPU Lockup Error Interrupt Status Flag)**

The LUST flag indicates a CPU Lockup error interrupt request. It is read-only and cleared by the NMICLR.LUCLR bit.

[Setting condition]

When the CPU Lockup error interrupt is generated.

[Clearing condition]

Be sure to clear the error status of the request source before clearing. If CPU does not clear the error status of the request source, the NMI status is set again even if this status is cleared, and an NMI request is issued to the CPU. If the CPU returns from the NMI handler, the CPU jumps back to the NMI handler.

In the case of level detection, use the following steps to clear the Status flag.

- (1) Negate the level of an input factor.
- (2) Perform a peripheral read access once and make sure the level interrupt is cleared.
- (3) Clear the status flag by NMICLR.LUSTCLR.

### 13.2.11 NMIER : Non-Maskable Interrupt Enable Register

Base address: ICU = 0x4000\_C000  
ICU\_NS = 0x5000\_C000

Offset address: 0x100

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LUEN	—	CMEN	BUSEN	—	—	—	—	NMIEN	OSTEN	—	—	PVD2EN	PVD1EN	WDTEN	IWDTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTEN	IWDT Underflow/Refresh Error Interrupt Enable 0: Disabled 1: Enabled	R/W <sup>*1 *2</sup>
1	WDTEN	WDT Underflow/Refresh Error Interrupt Enable 0: Disabled 1: Enabled	R/W <sup>*1 *2</sup>
2	PVD1EN	Voltage monitor 1 Interrupt Enable 0: Disabled 1: Enabled	R/W <sup>*1 *2</sup>
3	PVD2EN	Voltage monitor 2 Interrupt Enable 0: Disabled 1: Enabled	R/W <sup>*1 *2</sup>
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OSTEN	Main Clock Oscillation Stop Detection Interrupt Enable 0: Disabled 1: Enabled	R/W <sup>*1 *2</sup>
7	NMIEN	NMI Pin Interrupt Enable 0: Disabled 1: Enabled	R/W <sup>*1</sup>
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	BUSEN	Bus Error Interrupt Enable 0: Disabled 1: Enabled	R/W <sup>*1</sup>
13	CMEN	Common Memory Error Interrupt Enable 0: Disabled 1: Enabled	R/W <sup>*1</sup>
14	—	This bit is read as 0. The write value should be 0.	R/W
15	LUEN	CPU Lockup Error Interrupt Enable 0: Disabled 1: Enabled	R/W <sup>*1</sup>

Note: S-TYPE3, P-TYPE2

Note 1. You can write 1 to this bit only once after reset. Subsequent write accesses are invalid. Writing 0 to this bit is invalid.

Note 2. Do not write 1 to this bit when the source is used as an interrupt signal.

#### IWDTEN bit (IWDT Underflow/Refresh Error Interrupt Enable)

The IWDTEN bit enables IWDT underflow/refresh error interrupt as an NMI trigger.

#### WDTEN bit (WDT Underflow/Refresh Error Interrupt Enable)

The WDTEN bit enables WDT underflow/refresh error interrupt as an NMI trigger.

#### PVD1EN bit (Voltage monitor 1 Interrupt Enable)

The PVD1EN bit enables voltage monitor 1 interrupt as an NMI trigger.

**PVD2EN bit (Voltage monitor 2 Interrupt Enable)**

The PVD2EN bit enables voltage monitor 2 interrupt as an NMI trigger.

**OSTEN bit (Main Clock Oscillation Stop Detection Interrupt Enable)**

The OSTEN bit enables main clock oscillation stop detection interrupt as an NMI trigger.

**NMIEN bit (NMI Pin Interrupt Enable)**

The NMIEN bit enables NMI pin interrupt as an NMI trigger.

**BUSEN bit (Bus Error Interrupt Enable)**

The BUSEN bit enables bus error interrupt as an NMI trigger. Bus error includes MPU and TZF errors.

**CMEN bit (Common Memory Error Interrupt Enable)**

The CMEN bit enables common memory error interrupt as an NMI trigger. Common memory errors include SRAM Parity or Standby RAM Parity.

**LUEN bit (CPU Lockup Error Interrupt Enable)**

LUEN bit enables CPU Lockup error interrupt as an NMI trigger.

**13.2.12 NMICLR : Non-Maskable Interrupt Status Clear Register**

Base address: ICU = 0x4000\_C000  
ICU\_NS = 0x5000\_C000

Offset address: 0x110

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LUCL R	—	CMCL R	BUSC LR	—	—	—	—	NMICL R	OSTC LR	—	—	PVD2 CLR	PVD1 CLR	WDTC LR	IWDT CLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IWDTCLR	IWDT Underflow/Refresh Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.IWDTST flag	R/W <sup>1</sup>
1	WDTCLR	WDT Underflow/Refresh Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.WDTST flag	R/W <sup>1</sup>
2	PVD1CLR	Voltage Monitor 1 Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.PVD1ST flag	R/W <sup>1</sup>
3	PVD2CLR	Voltage Monitor 2 Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.PVD2ST flag.	R/W <sup>1</sup>
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	OSTCLR	Oscillation Stop Detection Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.OSTST flag	R/W <sup>1</sup>
7	NMICLR	NMI Pin Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.NMIST flag	R/W <sup>1</sup>
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	BUSCLR	Bus Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.BUSST flag	R/W <sup>1</sup>

Bit	Symbol	Function	R/W
13	CMCLR	Common Memory Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.CMST flag	R/W <sup>1</sup>
14	—	This bit is read as 0. The write value should be 0.	R/W
15	LUCLR	CPU Lockup Error Interrupt Status Flag Clear 0: No effect 1: Clear the NMISR.LUST flag	R/W <sup>1</sup>

Note: S-TYPE3, P-TYPE2

Note: here may be a difference in processing speed between the CPU and ICU, and the CPU may exit the interrupt handler before clearing the NMISR. Then the CPU will accidentally jump to the NMI handler again. To avoid this, be sure to read NMISR before exiting the NMI handler and make sure that NMISR is cleared before exiting the NMI handler.

Note 1. Only write 1 to this bit.

#### IWDTCLR bit (IWDT Underflow/Refresh Error Interrupt Status Flag Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. This bit is read as 0.

#### WDTCLR bit (WDT Underflow/Refresh Error Interrupt Status Flag Clear)

Writing 1 to the WDTCLR bit clears the NMISR.WDTST flag. This bit is read as 0.

#### PVD1CLR bit (Voltage Monitor 1 Interrupt Status Flag Clear)

Writing 1 to the PVD1CLR bit clears the NMISR.PVD1ST flag. This bit is read as 0.

#### PVD2CLR bit (Voltage Monitor 2 Interrupt Status Flag Clear)

Writing 1 to the PVD2CLR bit clears the NMISR.PVD2ST flag. This bit is read as 0.

#### OSTCLR bit (Oscillation Stop Detection Interrupt Status Flag Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. This bit is read as 0.

#### NMICLR bit (NMI Pin Interrupt Status Flag Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. This bit is read as 0.

#### BUSCLR bit (Bus Error Interrupt Status Flag Clear)

Writing 1 to the BUSCLR bit clears the NMISR.BUSST flag. This bit is read as 0.

#### CMCLR bit (Common Memory Error Interrupt Status Flag Clear)

Writing 1 to the CMCLR bit clears the NMISR.CMST flag. This bit is read as 0.

#### LUCLR bit (CPU Lockup Error Interrupt Status Flag Clear)

Writing 1 to the LUCLR bit clears the NMISR.LUST flag. This bit is read as 0.

### 13.2.13 NMICR : NMI Pin Interrupt Control Register

Base address: ICU\_COMMON = 0x4000\_6000  
ICU\_COMMON\_NS = 0x5000\_6000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	NFLTE N	—	NFCLKSEL[1:0]	—	—	—	—	NMIM D
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NMIMD	NMI Detection Set 0: Falling edge 1: Rising edge	R/W

Bit	Symbol	Function	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
5:4	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock Select 0 0: PCLKB 0 1: PCLKB/8 1 0: PCLKB/32 1 1: PCLKB/64	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	NFLTEN	NMI Digital Filter Enable 0: Disabled 1: Enabled	R/W

Note: S-TYPE3, P-TYPE2

Change the NMICR register settings before enabling NMI pin interrupts, that is, before setting NMIER.NMIEN to 1.

### NMIMD bit (NMI Detection Set)

The NMIMD bit selects the detection sensing method for the NMI pin interrupts.

### NFCLKSEL[1:0] bits (NMI Digital Filter Sampling Clock Select)

The NFCLKSEL[1:0] bits select the digital filter sampling clock for the NMI pin interrupts, selectable to:

- PCLKB (every cycle)
- PCLKB/8 (once every 8 cycles)
- PCLKB/32 (once every 32 cycles)
- PCLKB/64 (once every 64 cycles)

For details of the digital filter, see [section 13.5.6. Digital Filter](#).

### NFLTEN bit (NMI Digital Filter Enable)

The NFLTEN bit enables the digital filter used for NMI pin interrupts. The filter is enabled when NFLTEN is 1, and disabled when NFLTEN is 0. The NMI pin level is sampled at the clock cycle specified in NFCLKSEL[1:0]. When the sampled level matches three times, the output level from the digital filter changes. For details of the digital filter, see [section 13.5.6. Digital Filter](#).

## 13.2.14 IELSRn : ICU Event Link Setting Register n (n = 0 to 95)

Base address: ICU = 0x4000\_C000  
ICU\_NS = 0x5000\_C000

Offset address: 0x300 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	DTCE	—	—	—	—	—	—	—	IR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	IELS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	IELS[8:0]	ICU Event Link Select 0x00: Disable interrupts to the associated NVIC or DTC module Others: Event signal number to be linked. For details, see <a href="#">section 13.3.2. Event Number</a> .	R/W <sup>1</sup>
15:9	—	These bits are read as 0. The write value should be 0.	R/W



Bit	Symbol	Function	R/W
16	IR	Interrupt Status Flag 0: No interrupt request generated. 1: An interrupt request is generated.	R/W <sup>2</sup>
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	DTCE	DTC Activation Enable 0: DTC activation is disabled. 1: DTC activation is enabled.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note 1. The [15:0] bits in this register can only be accessed in halfword or word access. Byte access is ignored.

Note 2. Writing 1 to the IR flag is prohibited.

This register has different secure access permission depending on the setting of Trusted Event Route Control Register (TEVTRCR).  
If the security attribution is configured as secure,

- Secure access is allowed.
- Non-secure write access is ignored and non-secure read access is read as 0. TrustZone access error is generated.

If the security attribution is configured as non-secure and the trusted event route is disabled:

- Secure write access is ignored and secure read access is read as 0. TrustZone access error is generated.
- Non-secure access is allowed.

If the security attribution is configured as non-secure and the trusted event route is enabled:

- Secure access to IELS bit is allowed.
- Non-secure write access is ignored and non-secure read access is allowed.
- Secure write access to other bits is ignored and secure read access to other bits is read as 0.
- Non-secure access to other bits is allowed.
- TrustZone access error is not generated.

P-TYPE-2

The IELSRn register selects the interrupt source used by the NVIC. For details, see [Table 13.3](#). IELSRn corresponds to the NVIC interrupt input source number, where n = 0 to 95.

### IELS[8:0] bits (ICU Event Link Select)

The IELS[8:0] bits link an event signal to the associated NVIC or DTC module. Event options are classified into 8 groups (groups 0 to 7). For details, see [section 13.3. Vector Table](#). All IELS[8:0] bits must be written at the same time.

### IR flag (Interrupt Status Flag)

The IR status flag indicates an individual interrupt request from the event specified in IELS[8:0].

[Setting condition]

When an interrupt request is received from the associated peripheral module or IRQi pin.

[Clearing condition]

- The IR flag is cleared to 0 by writing 0.
- In the case of DTC.DISEL = 0. At the time other than the final transfer end in DTC transfer during DTCE = 1, IR flag repeat set and cleared by Hardware.
- In the case of DTC.DISEL = 1. For DTC transfers during DTCE = 1, the hardware does not clear the IR flag. Should be cleared by the CPU writing 0.

When DTC transfer except last transfer is completed (DTCE bit is changed from 1 to 0).

During DTCE = 1, write 0 to IR register is prohibited.

In the case of level detection, clear of the IR flag should follow the steps below.

1. Negate the input interrupt signal.
2. Run the peripheral read access once and wait for 2 clock cycles of the target module clock.
3. Clear the IR flag by writing 0.

Note: There may be a difference in processing speed between the CPU and ICU, and the CPU may exit the interrupt handler before clearing the IR. Then the CPU will accidentally jump to the interrupt handler again. To avoid this, be sure to read this register before exiting the interrupt handler and make sure that IR is cleared before exiting the interrupt handler.

**DTCE bit (DTC Activation Enable)**

When the DTCE bit is set to 1, the associated event is selected as the source for DTC activation.

[Setting condition]

- When 1 is written to the DTCE bit.

[Clearing condition]

- When the specified number of transfers is complete. For chain transfers, when the specified number of transfers for the last chain transfer is complete.
- When 0 is written to the DTCE bit.

Note: The secure attribution managed within the Arm CPU NVIC must match the SA (Security Attribution) of IELSEn (n = 0 to 95).

Note: Error during DTC transfer

If an error response occurs during DTC transfer, the DTC notifies the ICU that an error has occurred. ICU clears all bits of the target IELSRn (n = 0 to 95). IELSRn that is not the target is not cleared.

**13.2.15 WUPEN0 : Wake Up Interrupt Enable Register 0**

Base address: ICU = 0x4000\_C000  
ICU\_NS = 0x5000\_C000

Offset address: 0x1A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	IIC0WUPEN	AGT1CBWUPEN	AGT1CAWUPEN	AGT1UDWUPEN	USBF0WUPEN	—	RTCPRDWUPEN	RTCALMWUPEN	—	—	—	VBATTWUPEN	PVD2WUPEN	PVD1WUPEN	—	IWDTWUPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	IRQWUPEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	IRQWUPEN[15:0]	IRQn Interrupt Deep Sleep/Software Standby Mode Returns Enable(n = 0 to 15) 0: Deep Sleep/Software Standby Mode returns by IRQn interrupt is disabled . 1: Deep Sleep/Software Standby Mode returns by IRQn interrupt is enabled*1	R/W
16	IWDTWUPEN	IWDT Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by IWDT interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by IWDT interrupt is enabled	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W
18	PVD1WUPEN	PVD1 Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by PVD1 interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by PVD1 interrupt is enabled	R/W
19	PVD2WUPEN	PVD2 Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by PVD2 interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by PVD2 interrupt is enabled	R/W

Bit	Symbol	Function	R/W
20	VBATTWUPEN	VBATT Monitor Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by VBATT monitor interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by VBATT monitor interrupt is enabled	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
24	RTCALMWUPEN	RTC Alarm Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by RTC alarm interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by RTC alarm interrupt is enabled	R/W
25	RTCPRDWUPEN	RTC Period Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by RTC period interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by RTC period interrupt is enabled	R/W
26	—	This bit is read as 0. The write value should be 0.	R/W
27	USBFS0WUPEN	USBFS Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by USBFS interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by USBFS interrupt is enabled	R/W
28	AGT1UDWUPEN	AGT1 Underflow Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by AGT1 underflow interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by AGT1 underflow interrupt is enabled	R/W
29	AGT1CAWUPEN	AGT1 Compare Match A Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by AGT1 compare match A interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by AGT1 compare match A interrupt is enabled	R/W
30	AGT1CBWUPEN	AGT1 Compare Match B Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by AGT1 compare match B interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by AGT1 compare match B interrupt is enabled	R/W
31	IIC0WUPEN	IIC0 Address Match Interrupt Deep Sleep/Software Standby Mode Returns Enable 0: Deep Sleep/Software Standby Mode returns by IIC0 address match A interrupt is disabled 1: Deep Sleep/Software Standby Mode returns by IIC0 address match A interrupt is enabled	R/W

Note: S-TYPE4, P-TYPE2

Note 1. Description is a description of each bit.

### **IRQWUPEN[15:0] bits (IRQn Interrupt Deep Sleep/Software Standby Mode Returns Enable(n = 0 to 15))**

The IRQWUPEN[15:0] are the enable bits to control the use of the IRQn pin as a Deep Sleep/Software Standby mode return factor. n= 0 to 15

### **IWDTWUPEN bit (IWDT Interrupt Deep Sleep/Software Standby Mode Returns Enable)**

The IWDTWUPEN is the enable bit to control the use of the IWDT interrupt as a Deep Sleep/Software Standby mode return factor.

### **PVD1WUPEN bit (PVD1 Interrupt Deep Sleep/Software Standby Mode Returns Enable)**

The PVD1WUPEN is the enable bit to control the use of the PVD1 interrupt as a Deep Sleep/Software Standby mode return factor.

### **PVD2WUPEN bit (PVD2 Interrupt Deep Sleep/Software Standby Mode Returns Enable)**

The PVD2WUPEN is the enable bit to control the use of the PVD2 interrupt as a Deep Sleep/Software Standby mode return factor.

**VBATTWUPEN bit (VBATT Monitor Interrupt Deep Sleep/Software Standby Mode Returns Enable)**

The VBATTWUPEN is the enable bit to control the use of the VBATT monitor interrupt as a Deep Sleep/Software Standby mode return factor.

**RTCALMWUPEN bit (RTC Alarm Interrupt Deep Sleep/Software Standby Mode Returns Enable)**

The RTCALMWUPEN is the enable bit to control the use of the RTC alarm interrupt as a Deep Sleep/Software Standby mode return factor.

**RTCPRDWUPEN bit (RTC Period Interrupt Deep Sleep/Software Standby Mode Returns Enable)**

The RTCPRDWUPEN is the enable bit to control the use of the RTC period interrupt as a Deep Sleep/Software Standby mode return factor.

**USBFS0WUPEN bit (USBFS Interrupt Deep Sleep/Software Standby Mode Returns Enable)**

The USBFS0WUPEN is the enable bit to control the use of the USBFS interrupt as a Deep Sleep/Software Standby mode return factor.

**AGT1UDWUPEN bit (AGT1 Underflow Interrupt Deep Sleep/Software Standby Mode Returns Enable)**

The AGT1UDWUPEN is the enable bit to control the use of the AGT1 underflow interrupt as a Deep Sleep/Software Standby mode return factor.

**AGT1CAWUPEN bit (AGT1 Compare Match A Interrupt Deep Sleep/Software Standby Mode Returns Enable)**

The AGT1CAWUPEN is the enable bit to control the use of the AGT1 compare match A interrupt as a Deep Sleep/Software Standby mode return factor.

**AGT1CBWUPEN bit (AGT1 Compare Match B Interrupt Deep Sleep/Software Standby Mode Returns Enable)**

The AGT1CBWUPEN is the enable bit to control the use of the AGT1 compare match B interrupt as a Deep Sleep/Software Standby mode return factor.

**IIC0WUPEN bit (IIC0 Address Match Interrupt Deep Sleep/Software Standby Mode Returns Enable)**

The IIC0WUPEN is the enable bit to control the use of the IIC0 interrupt as a Deep Sleep/Software Standby mode return factor.

Note: The security attribution of this register is set for each wakeup event.

To avoid the occurrence of a security hole, the target event of a wakeup and the security attribution added to this bit must match.

**13.2.16 WUPEN1 : Wake Up interrupt enable register 1**

Base address: ICU = 0x4000\_C000  
ICU\_NS = 0x5000\_C000

Offset address: 0x1A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ULP1B WUPE N	ULP1A WUPE N	ULP1 UWUP EN	—	ULP0B WUPE N	ULP0A WUPE N	ULP0 UWUP EN	—	—	—	—	COMP HS0W UPEN	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
3	COMPHS0WUPEN	Comparator-HS0 Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by Comparator-HS0 interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT0 Underflow interrupt is enabled.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	ULP0UWUPEN	ULPT0 Underflow Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by ULPT0 interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT0 Underflow interrupt is enabled.	R/W
9	ULP0AWUPEN	ULPT0 Compare Match A Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by ULPT0 Compare match A interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT0 Compare match A interrupt is enabled.	R/W
10	ULP0BWUPEN	ULPT0 Compare Match B Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by ULPT0 Compare match B interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT0 Compare match B interrupt is enabled.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	ULP1UWUPEN	ULPT1 Underflow Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by ULPT1 Underflow interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT1 Underflow interrupt is enabled.	R/W
13	ULP1AWUPEN	ULPT1 Compare Match A Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by ULPT1 Compare match A interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT1 Compare match A interrupt is enabled.	R/W
14	ULP1BWUPEN	ULPT1 Compare Match B Interrupt Deep Sleep/Software Standby Mode returns Enable bit 0: Deep Sleep/Software Standby returns by ULPT1 Compare match B interrupt is disabled. 1: Deep Sleep/Software Standby returns by ULPT1 Compare match B interrupt is enabled.	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE4, P-TYPE2

#### **COMPHS0WUPEN bit (Comparator-HS0 Interrupt Deep Sleep/Software Standby Mode returns Enable bit)**

This bit is the enable bit to control the use of the Comparator-HS0 interrupt as a Deep Sleep/Software Standby return factor.

#### **ULP0UWUPEN bit (ULPT0 Underflow Interrupt Deep Sleep/Software Standby Mode returns Enable bit)**

This bit is the enable bit to control the use of the ULPT0 Underflow interrupt as a Deep Sleep/Software Standby return factor.

#### **ULP0AWUPEN bit (ULPT0 Compare Match A Interrupt Deep Sleep/Software Standby Mode returns Enable bit)**

This bit is the enable bit to control the use of the ULPT0 Compare match A interrupt as a Deep Sleep/Software Standby return factor.

#### **ULP0BWUPEN bit (ULPT0 Compare Match B Interrupt Deep Sleep/Software Standby Mode returns Enable bit)**

This bit is the enable bit to control the use of the ULPT0 Compare match B interrupt as a Deep Sleep/Software Standby return factor.

#### **ULP1UWUPEN bit (ULPT1 Underflow Interrupt Deep Sleep/Software Standby Mode returns Enable bit)**

This bit is the enable bit to control the use of the ULPT1 Underflow interrupt as a Deep Sleep/Software Standby return factor.

**ULP1AWUPEN bit (ULPT1 Compare Match A Interrupt Deep Sleep/Software Standby Mode returns Enable bit)**

This bit is the enable bit to control the use of the ULPT1 Compare match A interrupt as a Deep Sleep/Software Standby return factor.

**ULP1BWUPEN bit (ULPT1 Compare Match B Interrupt Deep Sleep/Software Standby Mode returns Enable bit)**

This bit is the enable bit to control the use of the ULPT1 Compare match B interrupt as a Deep Sleep/Software Standby return factor.

Note: The security attribution of this register is set for each wakeup event.

To avoid the occurrence of a security hole, the target event of a wakeup and the security attribution added to this bit must match.

### 13.3 Vector Table

The ICU detects maskable and non-maskable interrupts. Interrupt priorities are set up in the Arm NVIC. For information about these registers, see [section 13.10. Reference](#).

#### 13.3.1 Interrupt Vector Table

[Table 13.3](#) describes the interrupt vector table. The interrupt vector addresses conform to the NVIC specifications.

**Table 13.3** Interrupt vector table (1 of 4)

Exception number	IRQ number	Vector offset	Source	Description
0	—	0x000	Arm	Initial stack pointer
1	—	0x004	Arm	Initial program counter (reset vector)
2	—	0x008	Arm	Non-Maskable Interrupt (NMI)
3	—	0x00C	Arm	HardFault
4	—	0x010	Arm	MemManage Fault
5	—	0x014	Arm	BusFault
6	—	0x018	Arm	UsageFault
7	—	0x01C	Arm	SecureFault
8	—	0x020	Arm	Reserved
9	—	0x024	Arm	Reserved
10	—	0x028	Arm	Reserved
11	—	0x02C	Arm	Supervisor Call (SVCall)
12	—	0x030	Arm	DebugMonitor
13	—	0x034	Arm	Reserved
14	—	0x038	Arm	Pendable request for system service (PendableSrvReq)
15	—	0x03C	Arm	System Tick Timer (SysTick)
16	0	0x040	ICU.IELSR0	Event selected in the ICU.IELSR0 register
17	1	0x044	ICU.IELSR1	Event selected in the ICU.IELSR1 register
18	2	0x048	ICU.IELSR2	Event selected in the ICU.IELSR2 register
19	3	0x04C	ICU.IELSR3	Event selected in the ICU.IELSR3 register
20	4	0x050	ICU.IELSR4	Event selected in the ICU.IELSR4 register
21	5	0x054	ICU.IELSR5	Event selected in the ICU.IELSR5 register
22	6	0x058	ICU.IELSR6	Event selected in the ICU.IELSR6 register
23	7	0x05C	ICU.IELSR7	Event selected in the ICU.IELSR7 register

Table 13.3 Interrupt vector table (2 of 4)

Exception number	IRQ number	Vector offset	Source	Description
24	8	0x060	ICU.IELSR8	Event selected in the ICU.IELSR8 register
25	9	0x064	ICU.IELSR9	Event selected in the ICU.IELSR9 register
26	10	0x068	ICU.IELSR10	Event selected in the ICU.IELSR10 register
27	11	0x06C	ICU.IELSR11	Event selected in the ICU.IELSR11 register
28	12	0x070	ICU.IELSR12	Event selected in the ICU.IELSR12 register
29	13	0x074	ICU.IELSR13	Event selected in the ICU.IELSR13 register
30	14	0x078	ICU.IELSR14	Event selected in the ICU.IELSR14 register
31	15	0x07C	ICU.IELSR15	Event selected in the ICU.IELSR15 register
32	16	0x080	ICU.IELSR16	Event selected in the ICU.IELSR16 register
33	17	0x084	ICU.IELSR17	Event selected in the ICU.IELSR17 register
34	18	0x088	ICU.IELSR18	Event selected in the ICU.IELSR18 register
35	19	0x08C	ICU.IELSR19	Event selected in the ICU.IELSR19 register
36	20	0x090	ICU.IELSR20	Event selected in the ICU.IELSR20 register
37	21	0x094	ICU.IELSR21	Event selected in the ICU.IELSR21 register
38	22	0x098	ICU.IELSR22	Event selected in the ICU.IELSR22 register
39	23	0x09C	ICU.IELSR23	Event selected in the ICU.IELSR23 register
40	24	0x0A0	ICU.IELSR24	Event selected in the ICU.IELSR24 register
41	25	0x0A4	ICU.IELSR25	Event selected in the ICU.IELSR25 register
42	26	0x0A8	ICU.IELSR26	Event selected in the ICU.IELSR26 register
43	27	0x0AC	ICU.IELSR27	Event selected in the ICU.IELSR27 register
44	28	0x0B0	ICU.IELSR28	Event selected in the ICU.IELSR28 register
45	29	0x0B4	ICU.IELSR29	Event selected in the ICU.IELSR29 register
46	30	0x0B8	ICU.IELSR30	Event selected in the ICU.IELSR30 register
47	31	0x0BC	ICU.IELSR31	Event selected in the ICU.IELSR31 register
48	32	0x0C0	ICU.IELSR32	Event selected in the ICU.IELSR32 register
49	33	0x0C4	ICU.IELSR33	Event selected in the ICU.IELSR33 register
50	34	0x0C8	ICU.IELSR34	Event selected in the ICU.IELSR34 register
51	35	0x0CC	ICU.IELSR35	Event selected in the ICU.IELSR35 register
52	36	0x0D0	ICU.IELSR36	Event selected in the ICU.IELSR36 register
53	37	0x0D4	ICU.IELSR37	Event selected in the ICU.IELSR37 register
54	38	0x0D8	ICU.IELSR38	Event selected in the ICU.IELSR38 register
55	39	0x0DC	ICU.IELSR39	Event selected in the ICU.IELSR39 register
56	40	0x0E0	ICU.IELSR40	Event selected in the ICU.IELSR40 register
57	41	0x0E4	ICU.IELSR41	Event selected in the ICU.IELSR41 register
58	42	0x0E8	ICU.IELSR42	Event selected in the ICU.IELSR42 register
59	43	0x0EC	ICU.IELSR43	Event selected in the ICU.IELSR43 register
60	44	0x0F0	ICU.IELSR44	Event selected in the ICU.IELSR44 register
61	45	0x0F4	ICU.IELSR45	Event selected in the ICU.IELSR45 register
62	46	0x0F8	ICU.IELSR46	Event selected in the ICU.IELSR46 register
63	47	0x0FC	ICU.IELSR47	Event selected in the ICU.IELSR47 register
64	48	0x100	ICU.IELSR48	Event selected in the ICU.IELSR48 register

**Table 13.3** Interrupt vector table (3 of 4)

Exception number	IRQ number	Vector offset	Source	Description
65	49	0x104	ICU.IELSR49	Event selected in the ICU.IELSR49 register
66	50	0x108	ICU.IELSR50	Event selected in the ICU.IELSR50 register
67	51	0x10C	ICU.IELSR51	Event selected in the ICU.IELSR51 register
68	52	0x110	ICU.IELSR52	Event selected in the ICU.IELSR52 register
69	53	0x114	ICU.IELSR53	Event selected in the ICU.IELSR53 register
70	54	0x118	ICU.IELSR54	Event selected in the ICU.IELSR54 register
71	55	0x11C	ICU.IELSR55	Event selected in the ICU.IELSR55 register
72	56	0x120	ICU.IELSR56	Event selected in the ICU.IELSR56 register
73	57	0x124	ICU.IELSR57	Event selected in the ICU.IELSR57 register
74	58	0x128	ICU.IELSR58	Event selected in the ICU.IELSR58 register
75	59	0x12C	ICU.IELSR59	Event selected in the ICU.IELSR59 register
76	60	0x130	ICU.IELSR60	Event selected in the ICU.IELSR60 register
77	61	0x134	ICU.IELSR61	Event selected in the ICU.IELSR61 register
78	62	0x138	ICU.IELSR62	Event selected in the ICU.IELSR62 register
79	63	0x13C	ICU.IELSR63	Event selected in the ICU.IELSR63 register
80	64	0x140	ICU.IELSR64	Event selected in the ICU.IELSR64 register
81	65	0x144	ICU.IELSR65	Event selected in the ICU.IELSR65 register
82	66	0x148	ICU.IELSR66	Event selected in the ICU.IELSR66 register
83	67	0x14C	ICU.IELSR67	Event selected in the ICU.IELSR67 register
84	68	0x150	ICU.IELSR68	Event selected in the ICU.IELSR68 register
85	69	0x154	ICU.IELSR69	Event selected in the ICU.IELSR69 register
86	70	0x158	ICU.IELSR70	Event selected in the ICU.IELSR70 register
87	71	0x15C	ICU.IELSR71	Event selected in the ICU.IELSR71 register
88	72	0x160	ICU.IELSR72	Event selected in the ICU.IELSR72 register
89	73	0x164	ICU.IELSR73	Event selected in the ICU.IELSR73 register
90	74	0x168	ICU.IELSR74	Event selected in the ICU.IELSR74 register
91	75	0x16C	ICU.IELSR75	Event selected in the ICU.IELSR75 register
92	76	0x170	ICU.IELSR76	Event selected in the ICU.IELSR76 register
93	77	0x174	ICU.IELSR77	Event selected in the ICU.IELSR77 register
94	78	0x178	ICU.IELSR78	Event selected in the ICU.IELSR78 register
95	79	0x17C	ICU.IELSR79	Event selected in the ICU.IELSR79 register
96	80	0x180	ICU.IELSR80	Event selected in the ICU.IELSR80 register
97	81	0x184	ICU.IELSR81	Event selected in the ICU.IELSR81 register
98	82	0x188	ICU.IELSR82	Event selected in the ICU.IELSR82 register
99	83	0x18C	ICU.IELSR83	Event selected in the ICU.IELSR83 register
100	84	0x190	ICU.IELSR84	Event selected in the ICU.IELSR84 register
101	85	0x194	ICU.IELSR85	Event selected in the ICU.IELSR85 register
102	86	0x198	ICU.IELSR86	Event selected in the ICU.IELSR86 register
103	87	0x19C	ICU.IELSR87	Event selected in the ICU.IELSR87 register
104	88	0x1A0	ICU.IELSR88	Event selected in the ICU.IELSR88 register
105	89	0x1A4	ICU.IELSR89	Event selected in the ICU.IELSR89 register



**Table 13.3** Interrupt vector table (4 of 4)

Exception number	IRQ number	Vector offset	Source	Description
106	90	0x1A8	ICU.IELSR90	Event selected in the ICU.IELSR90 register
107	91	0x1AC	ICU.IELSR91	Event selected in the ICU.IELSR91 register
108	92	0x1B0	ICU.IELSR92	Event selected in the ICU.IELSR92 register
109	93	0x1B4	ICU.IELSR93	Event selected in the ICU.IELSR93 register
110	94	0x1B8	ICU.IELSR94	Event selected in the ICU.IELSR94 register
111	95	0x1BC	ICU.IELSR95	Event selected in the ICU.IELSR95 register

### 13.3.2 Event Number

The following table lists heading details for [Table 13.4](#), which describes each event number.

Heading	Description
Interrupt request source	Name of the source generating the interrupt request
Name	Name of the interrupt
Connect to NVIC	“✓” indicates the interrupt can be used as a CPU interrupt
Invoke DTC	“✓” indicates the interrupt can be used to request DTC activation
Invoke DMAC	“✓” indicates the interrupt can be used to request DMAC activation
Canceling CPU Deep Sleep	“✓” indicates the interrupt can be used to request a return from CPU Deep Sleep mode
Canceling Software Standby	“✓” indicates the interrupt can be used to request a return from Software Standby mode
Canceling Deep Software Standby	“✓” indicates the interrupt can be used to request a return from Deep Software Standby mode

**Table 13.4** Event table (1 of 8)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x001	Port	PORT_IRQ0	✓	✓	✓	✓	✓	✓
0x002		PORT_IRQ1	✓	✓	✓	✓	✓	✓
0x003		PORT_IRQ2	✓	✓	✓	✓	✓	✓
0x004		PORT_IRQ3	✓	✓	✓	✓	✓	✓
0x005		PORT_IRQ4	✓	✓	✓	✓	✓	✓
0x006		PORT_IRQ5	✓	✓	✓	✓	✓	✓
0x007		PORT_IRQ6	✓	✓	✓	✓	✓	✓
0x008		PORT_IRQ7	✓	✓	✓	✓	✓	✓
0x009		PORT_IRQ8	✓	✓	✓	✓	✓	✓
0x00A		PORT_IRQ9	✓	✓	✓	✓	✓	✓
0x00B		PORT_IRQ10	✓	✓	✓	✓	✓	✓
0x00C		PORT_IRQ11	✓	✓	✓	✓	✓	✓
0x00D		PORT_IRQ12	✓	✓	✓	✓	✓	✓
0x00E		PORT_IRQ13	✓	✓	✓	✓	✓	✓
0x00F		PORT_IRQ14	✓	✓	✓	✓	✓	✓
0x010		PORT_IRQ15	✓	✓	✓	✓	✓	✓
0x011	DMAC00	DMAC00_INT	✓	✓	—	—	—	—
0x012	DMAC01	DMAC01_INT	✓	✓	—	—	—	—

Table 13.4 Event table (2 of 8)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x013	DMAC02	DMAC02_INT	✓	✓	—	—	—	—
0x014	DMAC03	DMAC03_INT	✓	✓	—	—	—	—
0x015	DMAC04	DMAC04_INT	✓	✓	—	—	—	—
0x016	DMAC05	DMAC05_INT	✓	✓	—	—	—	—
0x017	DMAC06	DMAC06_INT	✓	✓	—	—	—	—
0x018	DMAC07	DMAC07_INT	✓	✓	—	—	—	—
0x022	DTC0	DTC0_COMPLETE	✓	—	—	—	—	—
0x027	DMAC/DTC	DMA0_TRANSERR	✓	—	—	—	—	—
0x029	DBG	DBG_CTIIRQ0	✓	—	—	—	—	—
0x02A		DBG_CTIIRQ1	✓	—	—	—	—	—
0x02B		DBG_JBRXI	✓	—	—	—	—	—
0x030	FCU	FCU_FIFERR	✓	—	—	—	—	—
0x031		FCU_FRDYI	✓	—	—	—	—	—
0x038	PVD1	PVD_PVD1	✓	—	—	✓	✓	✓ <sup>*4</sup>
0x039	PVD2	PVD_PVD2	✓	—	—	✓	✓	✓ <sup>*4</sup>
0x03D	BBF	VBATT_TADI	✓	—	—	✓	✓	✓
0x03E	MOSC	MOSC_STOP	✓	—	—	—	—	—
0x040	ULPT0	ULPT0_ULPTI	✓	✓	✓	✓	✓	✓ <sup>*5</sup>
0x041		ULPT0_ULPTCMAI	✓	✓	✓	✓	✓	—
0x042		ULPT0_ULPTCMBI	✓	✓	✓	✓	✓	—
0x043	ULPT1	ULPT1_ULPTI	✓	✓	✓	✓	✓	✓ <sup>*5</sup>
0x044		ULPT1_ULPTCMAI	✓	✓	✓	✓	✓	—
0x045		ULPT1_ULPTCMBI	✓	✓	✓	✓	✓	—
0x046	AGT0	AGT0_AGTI	✓	✓	✓	—	—	—
0x047		AGT0_AGTICMAI	✓	✓	✓	—	—	—
0x048		AGT0_AGTICMBI	✓	✓	✓	—	—	—
0x049	AGT1	AGT1_AGTI	✓	✓	✓	✓	✓	—
0x04A		AGT1_AGTICMAI	✓	✓	✓	✓	✓	—
0x04B		AGT1_AGTICMBI	✓	✓	✓	✓	✓	—
0x052	IWDT	IWDT_NMIUNDF	✓	—	—	✓	✓	✓ <sup>*5</sup>
0x053	WDT	WDT0_NMIUNDF	✓	—	—	—	—	—
0x055	RTC	RTC_ALM	✓	—	—	✓	✓	✓
0x056		RTC_PRD	✓	—	—	✓	✓	✓
0x057		RTC_CUP	✓	—	—	—	—	—
0x058	USBFS	USBFS_D0FIFO	✓	✓	✓	—	—	—
0x059		USBFS_D1FIFO	✓	✓	✓	—	—	—
0x05A		USBFS_USBI	✓	—	—	—	—	—
0x05B		USBFS_USBR	✓	—	—	✓	✓	✓ <sup>*5</sup>

Table 13.4 Event table (3 of 8)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x05C	IIC0	IIC0_RXI	✓	✓	✓	—	—	—
0x05D		IIC0_TXI	✓	✓	✓	—	—	—
0x05E		IIC0_TEI	✓	—	—	—	—	—
0x05F		IIC0_EEI	✓	—	—	—	—	—
0x060		IIC0_WUI	✓	—	—	✓	✓	—
0x061	IIC1	IIC1_RXI	✓	✓	✓	—	—	—
0x062		IIC1_TXI	✓	✓	✓	—	—	—
0x063		IIC1_TEI	✓	—	—	—	—	—
0x064		IIC1_EEI	✓	—	—	—	—	—
0x073	SSI0	SSI0_SSITXI	✓	✓	✓	—	—	—
0x074		SSI0_SSIRXI	✓	✓	✓	—	—	—
0x076		SSI0_SSIF	✓	—	—	—	—	—
0x079	SSI1	SSI1_SSIRT	✓	✓	✓	—	—	—
0x07A		SSI1_SSIF	✓	—	—	—	—	—
0x07B	ACMPHS	ACMP_HS0	✓	—	—	✓*1	✓*1	—
0x07C		ACMP_HS1	✓	—	—	—	—	—
0x083	ELC	ELC_SWEVT0	✓*3	✓	—	—	—	—
0x084		ELC_SWEVT1	✓*3	✓	—	—	—	—
0x088	PORT	IOPORT_GROUP1	✓	✓*2	✓*2	—	—	—
0x089		IOPORT_GROUP2	✓	✓*2	✓*2	—	—	—
0x08A		IOPORT_GROUP3	✓	✓*2	✓*2	—	—	—
0x08B		IOPORT_GROUP4	✓	✓*2	✓*2	—	—	—
0x08C	CAC	CAC_FEERI	✓	—	—	—	—	—
0x08D		CAC_MENDI	✓	—	—	—	—	—
0x08E		CAC_OVFI	✓	—	—	—	—	—
0x08F	POEG	POEG_GROUPA	✓	—	—	—	—	—
0x090		POEG_GROUPB	✓	—	—	—	—	—
0x091		POEG_GROUPC	✓	—	—	—	—	—
0x092		POEG_GROUPD	✓	—	—	—	—	—
0x0A1	GPT0	GPT0_CCMPA	✓	✓	✓	—	—	—
0x0A2		GPT0_CCMPB	✓	✓	✓	—	—	—
0x0A3		GPT0_CMPC	✓	✓	✓	—	—	—
0x0A4		GPT0_CMPD	✓	✓	✓	—	—	—
0x0A5		GPT0_CMPE	✓	✓	✓	—	—	—
0x0A6		GPT0_CMPF	✓	✓	✓	—	—	—
0x0A7		GPT0_OVF	✓	✓	✓	—	—	—
0x0A8		GPT0_UDF	✓	✓	✓	—	—	—
0x0A9		GPT0_PC	✓	✓	✓	—	—	—

Table 13.4 Event table (4 of 8)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x0AA	GPT1	GPT1_CCMPA	✓	✓	✓	—	—	—
0x0AB		GPT1_CCMPB	✓	✓	✓	—	—	—
0x0AC		GPT1_CMPC	✓	✓	✓	—	—	—
0x0AD		GPT1_CMPD	✓	✓	✓	—	—	—
0x0AE		GPT1_CMPE	✓	✓	✓	—	—	—
0x0AF		GPT1_CMPF	✓	✓	✓	—	—	—
0x0B0		GPT1_OVF	✓	✓	✓	—	—	—
0x0B1		GPT1_UDF	✓	✓	✓	—	—	—
0x0B2		GPT1_PC	✓	✓	✓	—	—	—
0x0B3		GPT2	GPT2_CCMPA	✓	✓	✓	—	—
0x0B4	GPT2_CCMPB		✓	✓	✓	—	—	—
0x0B5	GPT2_CMPC		✓	✓	✓	—	—	—
0x0B6	GPT2_CMPD		✓	✓	✓	—	—	—
0x0B7	GPT2_CMPE		✓	✓	✓	—	—	—
0x0B8	GPT2_CMPF		✓	✓	✓	—	—	—
0x0B9	GPT2_OVF		✓	✓	✓	—	—	—
0x0BA	GPT2_UDF		✓	✓	✓	—	—	—
0x0BB	GPT2_PC		✓	✓	✓	—	—	—
0x0BC	GPT3		GPT3_CCMPA	✓	✓	✓	—	—
0x0BD		GPT3_CCMPB	✓	✓	✓	—	—	—
0x0BE		GPT3_CMPC	✓	✓	✓	—	—	—
0x0BF		GPT3_CMPD	✓	✓	✓	—	—	—
0x0C0		GPT3_CMPE	✓	✓	✓	—	—	—
0x0C1		GPT3_CMPF	✓	✓	✓	—	—	—
0x0C2		GPT3_OVF	✓	✓	✓	—	—	—
0x0C3		GPT3_UDF	✓	✓	✓	—	—	—
0x0C4		GPT3_PC	✓	✓	✓	—	—	—
0x0C5	GPT4	GPT4_CCMPA	✓	✓	✓	—	—	—
0x0C6		GPT4_CCMPB	✓	✓	✓	—	—	—
0x0C7		GPT4_CMPC	✓	✓	✓	—	—	—
0x0C8		GPT4_CMPD	✓	✓	✓	—	—	—
0x0C9		GPT4_CMPE	✓	✓	✓	—	—	—
0x0CA		GPT4_CMPF	✓	✓	✓	—	—	—
0x0CB		GPT4_OVF	✓	✓	✓	—	—	—
0x0CC		GPT4_UDF	✓	✓	✓	—	—	—

Table 13.4 Event table (5 of 8)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x0CE	GPT5	GPT5_CCMPA	✓	✓	✓	—	—	—
0x0CF		GPT5_CCMPB	✓	✓	✓	—	—	—
0x0D0		GPT5_CMPC	✓	✓	✓	—	—	—
0x0D1		GPT5_CMPD	✓	✓	✓	—	—	—
0x0D2		GPT5_CMPE	✓	✓	✓	—	—	—
0x0D3		GPT5_CMPF	✓	✓	✓	—	—	—
0x0D4		GPT5_OVF	✓	✓	✓	—	—	—
0x0D5		GPT5_UDF	✓	✓	✓	—	—	—
0x0FB	GPT10	GPT10_CCMPA	✓	✓	✓	—	—	—
0x0FC		GPT10_CCMPB	✓	✓	✓	—	—	—
0x0FD		GPT10_CMPC	✓	✓	✓	—	—	—
0x0FE		GPT10_CMPD	✓	✓	✓	—	—	—
0x0FF		GPT10_CMPE	✓	✓	✓	—	—	—
0x100		GPT10_CMPF	✓	✓	✓	—	—	—
0x101		GPT10_OVF	✓	✓	✓	—	—	—
0x102		GPT10_UDF	✓	✓	✓	—	—	—
0x103	GPT10_PC	✓	✓	✓	—	—	—	
0x104	GPT11	GPT11_CCMPA	✓	✓	✓	—	—	—
0x105		GPT11_CCMPB	✓	✓	✓	—	—	—
0x106		GPT11_CMPC	✓	✓	✓	—	—	—
0x107		GPT11_CMPD	✓	✓	✓	—	—	—
0x108		GPT11_CMPE	✓	✓	✓	—	—	—
0x109		GPT11_CMPF	✓	✓	✓	—	—	—
0x10A		GPT11_OVF	✓	✓	✓	—	—	—
0x10B		GPT11_UDF	✓	✓	✓	—	—	—
0x10D	GPT12	GPT12_CCMPA	✓	✓	✓	—	—	—
0x10E		GPT12_CCMPB	✓	✓	✓	—	—	—
0x10F		GPT12_CMPC	✓	✓	✓	—	—	—
0x110		GPT12_CMPD	✓	✓	✓	—	—	—
0x111		GPT12_CMPE	✓	✓	✓	—	—	—
0x112		GPT12_CMPF	✓	✓	✓	—	—	—
0x113		GPT12_OVF	✓	✓	✓	—	—	—
0x114		GPT12_UDF	✓	✓	✓	—	—	—

Table 13.4 Event table (6 of 8)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x116	GPT13	GPT13_CCMPA	✓	✓	✓	—	—	—
0x117		GPT13_CCMPB	✓	✓	✓	—	—	—
0x118		GPT13_CMPC	✓	✓	✓	—	—	—
0x119		GPT13_CMPD	✓	✓	✓	—	—	—
0x11A		GPT13_CMPE	✓	✓	✓	—	—	—
0x11B		GPT13_CMPF	✓	✓	✓	—	—	—
0x11C		GPT13_OVF	✓	✓	✓	—	—	—
0x11D		GPT13_UDF	✓	✓	✓	—	—	—
0x120	Ether	ETHER_EINT0	✓	—	—	—	—	—
0x124	SCI0	SCI0_RXI	✓	✓	✓	—	—	—
0x125		SCI0_TXI	✓	✓	✓	—	—	—
0x126		SCI0_TEI	✓	—	—	—	—	—
0x127		SCI0_ERI	✓	—	—	—	—	—
0x128		SCI0_AED	✓	—	—	—	—	—
0x129		SCI0_BFD	✓	—	—	—	—	—
0x12A		SCI0_AM	✓	—	—	—	—	—
0x12B		SCI1	SCI1_RXI	✓	✓	✓	—	—
0x12C	SCI1_TXI		✓	✓	✓	—	—	—
0x12D	SCI1_TEI		✓	—	—	—	—	—
0x12E	SCI1_ERI		✓	—	—	—	—	—
0x12F	SCI1_AED		✓	—	—	—	—	—
0x130	SCI1_BFD		✓	—	—	—	—	—
0x131	SCI1_AM		✓	—	—	—	—	—
0x132	SCI2	SCI2_RXI	✓	✓	✓	—	—	—
0x133		SCI2_TXI	✓	✓	✓	—	—	—
0x134		SCI2_TEI	✓	—	—	—	—	—
0x135		SCI2_ERI	✓	—	—	—	—	—
0x138		SCI2_AM	✓	—	—	—	—	—
0x139	SCI3	SCI3_RXI	✓	✓	✓	—	—	—
0x13A		SCI3_TXI	✓	✓	✓	—	—	—
0x13B		SCI3_TEI	✓	—	—	—	—	—
0x13C		SCI3_ERI	✓	—	—	—	—	—
0x13F		SCI3_AM	✓	—	—	—	—	—
0x140	SCI4	SCI4_RXI	✓	✓	✓	—	—	—
0x141		SCI4_TXI	✓	✓	✓	—	—	—
0x142		SCI4_TEI	✓	—	—	—	—	—
0x143		SCI4_ERI	✓	—	—	—	—	—
0x146		SCI4_AM	✓	—	—	—	—	—

Table 13.4 Event table (7 of 8)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x163	SCI9	SCI9_RXI	✓	✓	✓	—	—	—
0x164		SCI9_TXI	✓	✓	✓	—	—	—
0x165		SCI9_TEI	✓	—	—	—	—	—
0x166		SCI9_ERI	✓	—	—	—	—	—
0x169		SCI9_AM	✓	—	—	—	—	—
0x178	SPI0	SPI0_SPRI	✓	✓	✓	—	—	—
0x179		SPI0_SPTI	✓	✓	✓	—	—	—
0x17A		SPI0_SPII	✓	—	—	—	—	—
0x17B		SPI0_SPEI	✓	—	—	—	—	—
0x17C		SPI0_SPCEND	✓	—	—	—	—	—
0x17D	SPI1	SPI1_SPRI	✓	✓	✓	—	—	—
0x17E		SPI1_SPTI	✓	✓	✓	—	—	—
0x17F		SPI1_SPII	✓	—	—	—	—	—
0x180		SPI1_SPEI	✓	—	—	—	—	—
0x181		SPI1_SPCEND	✓	—	—	—	—	—
0x182	OSPI0	OSPI0_ERR	✓	—	—	—	—	—
0x183		OSPI0_CMP	✓	—	—	—	—	—
0x185	CANFD	CAN_RXF	✓	—	—	—	—	—
0x186		CAN_GLERR	✓	—	—	—	—	—
0x187		CAN0_RF_DMARE Q0	✓	✓	✓	—	—	—
0x188		CAN0_RF_DMARE Q1	✓	✓	✓	—	—	—
0x18B		CAN1_RF_DMARE Q0	✓	✓	✓	—	—	—
0x18C		CAN1_RF_DMARE Q1	✓	✓	✓	—	—	—
0x18F		CAN0_TX	✓	—	—	—	—	—
0x190		CAN0_CHERR	✓	—	—	—	—	—
0x191		CAN0_COMFRX	✓	—	—	—	—	—
0x192		CAN0_CF_DMARE Q	✓	✓	✓	—	—	—
0x193		CAN0_RXMB	✓	—	—	—	—	—
0x194		CAN1_TX	✓	—	—	—	—	—
0x195		CAN1_CHERR	✓	—	—	—	—	—
0x196		CAN1_COMFRX	✓	—	—	—	—	—
0x197		CAN1_CF_DMARE Q	✓	✓	✓	—	—	—
0x198		CAN1_RXMB	✓	—	—	—	—	—
0x19B		CAN0_MRAM_ERI	✓	—	—	—	—	—
0x19C	CAN1_MRAM_ERI	✓	—	—	—	—	—	

**Table 13.4** Event table (8 of 8)

Event number	Interrupt request source	Name	IELSRn		DELSRn	Canceling CPU Deep Sleep	Canceling Software Standby	Canceling Deep Software Standby
			Connect to NVIC	Invoke DTC	Invoke DMAC			
0x1AE	ADC120	ADC120_ADI	✓	✓	✓	—	—	—
0x1AF		ADC120_GBADI	✓	✓	✓	—	—	—
0x1B0		ADC120_CMPAI	✓	—	—	—	—	—
0x1B1		ADC120_CMPBI	✓	—	—	—	—	—
0x1B2		ADC120_WCMPPM	—	✓	✓	—	—	—
0x1B3		ADC120_WCMPPUM	—	✓	✓	—	—	—
0x1B4	ADC121	ADC121_ADI	✓	✓	✓	—	—	—
0x1B5		ADC121_GBADI	✓	✓	✓	—	—	—
0x1B6		ADC121_CMPAI	✓	—	—	—	—	—
0x1B7		ADC121_CMPBI	✓	—	—	—	—	—
0x1B8		ADC121_WCMPPM	—	✓	✓	—	—	—
0x1B9		ADC121_WCMPPUM	—	✓	✓	—	—	—
0x1BA	DOC	DOC_DOPCI	✓	—	—	—	—	—
0x1BC	RSIP-E51A	RSIP_TADI	✓	—	—	—	—	—
0x1DA	CEU	CEU_CEUI	✓	—	—	—	—	—

Note 1. Only supported when CMPCTL.CSTEN = 1.

Note 2. Only the first edge detection is valid.

Note 3. Support only interrupt after DTC transfer.

Note 4. Only supported in Deep Software Standby mode 1 and Deep Software Standby mode 2.

Note 5. Only supported in Deep Software Standby mode 1.

## 13.4 Maskable Interrupt Operation

The ICU performs the following functions:

- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations such as CPU interrupt, DTC activation.

### 13.4.1 Detecting Interrupts

The ICU selects an event source input from a peripheral function interrupt or an external pin interrupt with IELSRn.IELS [8:0].

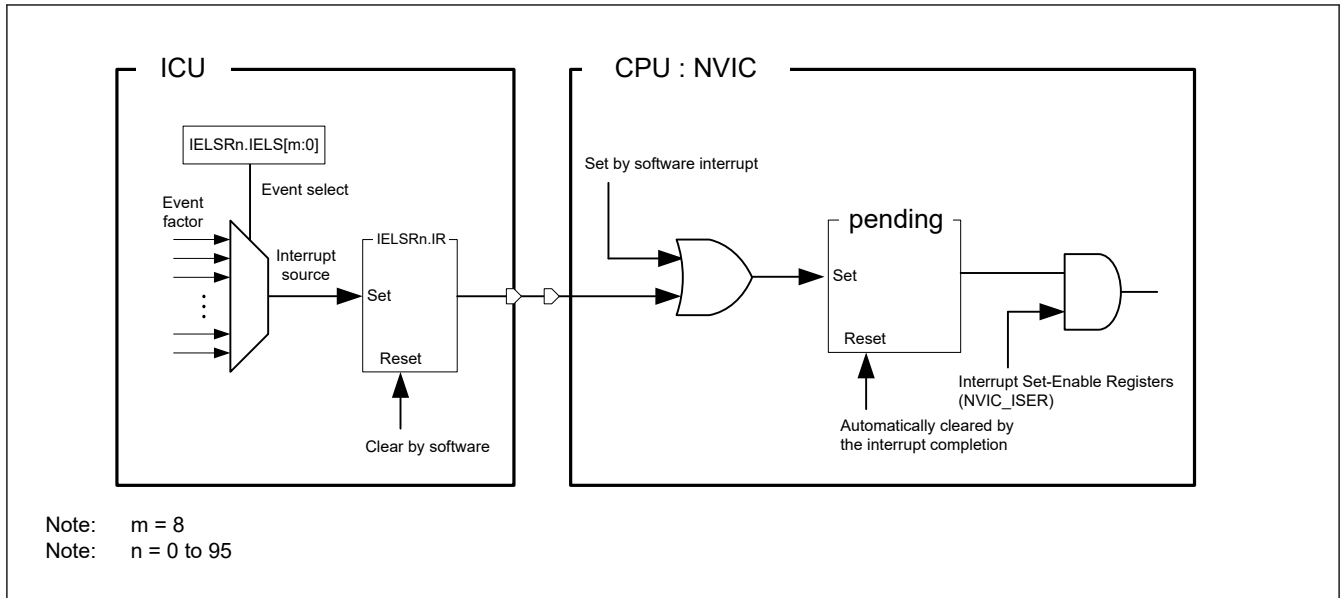
The accepted interrupt source sets the IELSRn.IR to 1 and sends an interrupt request to the NVIC.

External pin interrupt requests are detected by either:

- Edges (falling edge, rising edge, or rising and falling edges)
- Level (low level) of the interrupt signal.

Set the IRQCRi.IRQMD[1:0] bits to select the detection mode for the IRQi pins. For interrupt sources associated with peripheral modules, see [Table 13.3](#) and [Table 13.4](#). Events must be accepted by the NVIC before an interrupt occurs and is accepted by the CPU.





**Figure 13.2** Interrupt path of the ICU and CPU (NVIC)

## 13.5 Maskable Interrupt setting procedure

### 13.5.1 Operations During an Interrupt

-When an interrupt is generated: except for software interrupts:

The IELSRn.IR ( $n = 0$  to 95) flag and Interrupt Set Pending Register (NVIC\_ISPRn) are set.

- When an interrupt is generated: software interrupts:

Interrupt Set Pending Register (NVIC\_ISPRn) are set.

- When an interrupt is complete:

Clear the IELSRn.IR ( $n = 0$  to 95) flag by software. The Interrupt Set Pending Register (NVIC\_ISPRn) is automatically cleared.

**Note:** There may be a difference in processing speed between the CPU and ICU, and the CPU may exit the interrupt handler before clearing the IR. Then the CPU will accidentally jump to the interrupt handler again. To avoid this, be sure to read IELSRn ( $n = 0$  to 95) register before exiting the interrupt handler and make sure that IR is cleared before exiting the interrupt handler.

### 13.5.2 Enabling Interrupt Requests

The procedure for enabling an interrupt request is as follows:

1. Set the Interrupt Set-Enable register (NVIC\_ISER).
2. Set the IELSRn.IELS[8:0] bits as the interrupt source.
3. Specify the operation settings for the event source, such as Deep Sleep/Software Standby mode cancellation (WUPEN register setting).

### 13.5.3 Disabling Interrupt Requests

The procedure to disable the interrupt request is as follows:

1. Disable the operation settings for the event source, such as Deep Sleep/Software Standby mode cancellation (WUPEN register setting).
2. Clear the interrupt source setting (IELSRn.IELS[8:0] = 0x00).
3. Clear the interrupt status flag (IELSRn.IR = 0).
4. Clear the interrupt Clear-Enable register (NVIC\_ICER) and interrupt Clear-Pending register (NVIC\_ICPR).

### 13.5.4 Polling for interrupts

The procedure for polling for interrupt requests is as follows:

1. Set the Interrupt Clear-Enable register (NVIC\_ICER).
2. Set the IELSRn.IELS[8:0] bits as the interrupt source.
3. Specify the operation settings for the event source, such as DMAC activation (DELSRn.DELS[8:0]), Deep Sleep/ Software Standby mode cancellation (WUPEN register setting).
4. Poll the interrupt Set-Pending register (NVIC\_ISPR).

### 13.5.5 Selecting Interrupt Request Destinations

The available destinations are fixed for each interrupt, as described in [Table 13.3](#), [Table 13.4](#).

The interrupt output destination, CPU and DTC can be independently selected for each interrupt source.

Use an interrupt request destination setting that is indicated by a “✓” in the event list (see [section 13.3.2. Event Number](#)).

Note: Setting the same interrupt source for IELSRn and DELSRn is prohibited.

If the DMAC or DTC is selected as the destination for requests from an IRQi pin, you must set the IRQCRi.IRQMD[1:0] bits for that interrupt to select edge detection.

#### 13.5.5.1 CPU interrupt request

When IELSRn.DTCE = 0, the event specified in the IELSRn register is output to the NVIC.

#### 13.5.5.2 DTC activation

When IELSRn.DTCE = 1, the event specified in the IELSRn register is output to the DTC. Use the following procedure:

1. Set the IELSRn.IELS[8:0] bits to the target event and set the IELSRn.DTCE bit to 1.
2. Set the DTC Module Start bit (DTCST.DTCST) to 1.

[Table 13.5](#) shows operation when the DTC is the interrupt request destination.

**Table 13.5 Operation when DTC becomes interrupt request destination**

Interrupt request destination	DISEL*1	Remaining transfer operations	Operation per request	IR*2	Interrupt request destination after transfer
DTC*3	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	CPU (IELSRn.DTCE bit is automatically cleared)
	0	≠ 0	DTC transfer	Cleared at the start of DTC data transfer after reading DTC transfer data	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	CPU (IELSRn.DTCE bit is automatically cleared)

Note 1. DTC.MRB.DISEL bit controls the interrupt generates timing from DTC to CPU.

Note 2. When the IELSRn.IR flag is 1, an interrupt request (DTC activation request) that occurs again is ignored.

Note 3. For chain transfers, DTC transfer continues until the last chain transfer ends. The DISEL bit state and the remaining transfer count determine whether a CPU interrupt occurs, the IELSRn.IR flag clear timing, and the interrupt request destination after transfer. See [Table 17.2](#) in [section 17, Data Transfer Controller \(DTC\)](#).

Note: Error during DTC transfer

If an error response occurs during DTC transfer, the DTC notifies the ICU that an error has occurred. ICU clears all bits of the target IELSRn (n = 0 to 95). IELSRn that is not the target is not cleared.

### 13.5.6 Digital Filter

A digital filter function is provided for the external interrupt request pins  $IRQ_i$ , ( $i = 0$  to  $15$ ) and the NMI pin interrupt. It samples input signals on the filter PCLKB sampling clock and removes any signal with a pulse width less than 3 sampling cycles.

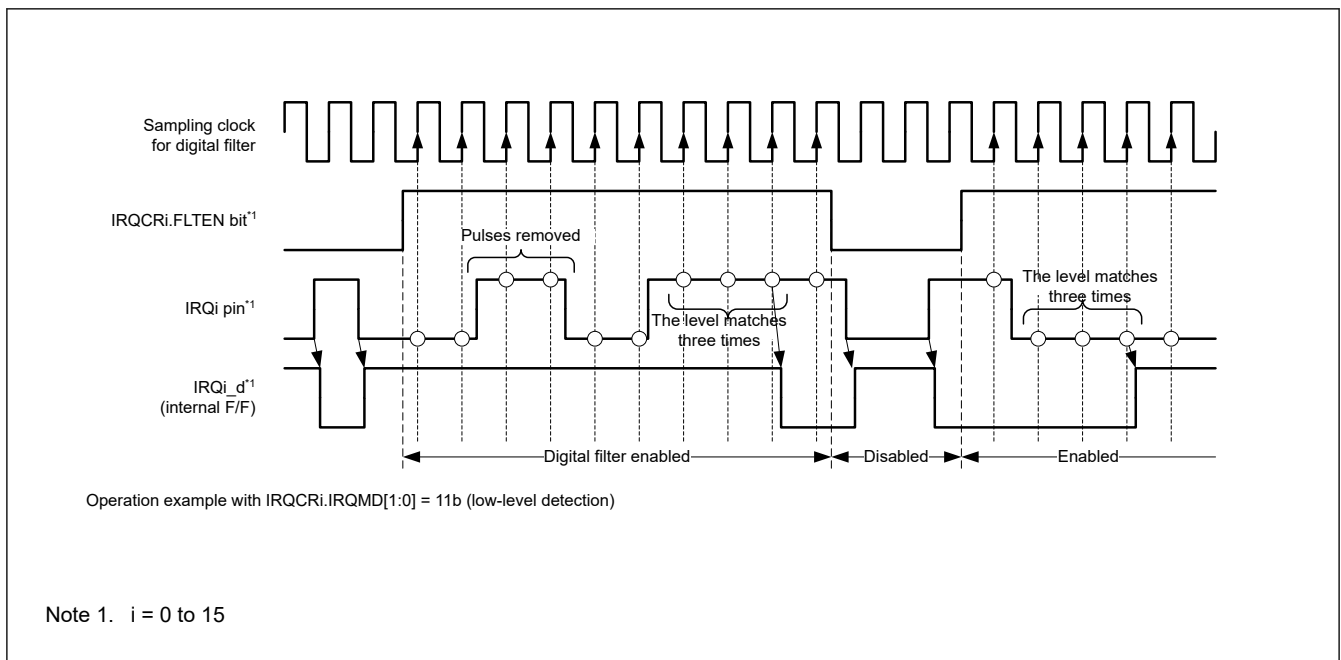
To use the digital filter for an  $IRQ_i$  pin:

1. Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the  $IRQCR_i.FCLKSEL[1:0]$  bits ( $i = 0$  to  $15$ ).
2. Set the  $IRQCR_i.FLTEN$  bit ( $i = 0$  to  $15$ ) to 1 (digital filter enabled).

To use the digital filter for an NMI pin:

1. Set the sampling clock cycle to PCLKB, PCLKB/8, PCLKB/32, or PCLKB/64 in the  $NMICR.NFCLKSEL[1:0]$  bits.
2. Set the  $NMICR.NFLTEN$  bit to 1 (digital filter enabled).

Figure 13.3 shows an example of digital filter operation.



**Figure 13.3 Digital filter operation example**

During Software Standby mode or Deep Software Standby mode, the digital filter is forcibly turned off by Hardware. After returning to normal mode from Software Standby mode or Deep Software Standby mode, follow the values of  $IRQCR_i.FLTEN$  ( $i = 0$  to  $15$ ) and  $NMICR.NFLTEN$ . Once the filter is disabled, the event information sampled up to that point will be lost.

### 13.5.7 External Pin Interrupts

To use external pin interrupts: Configure I/O ports settings.

1. Clear the  $IRQCR_i.FLTEN$  bit ( $i = 0$  to  $15$ ) to 0 (digital filter disabled).
2. Specify or confirm the I/O port settings.
3. Set the  $IRQMD[1:0]$  the  $FCLKSEL[1:0]$  bits, and the  $FLTEN$  bit of the  $IRQCR_i$  register ( $i = 0$  to  $15$ ) register.
4. Select the IRQ pin as follows:
  - If the IRQ pin is to be used for CPU interrupt requests, set the  $IELSR_n.IELS[8:0]$  ( $n = 0$  to  $95$ ) bits and the  $IELSR_n.DTCE$  ( $n = 0$  to  $95$ ) bit to 0.
  - If the IRQ pin is to be used for DTC activation, set the  $IELSR_n.IELS[8:0]$  ( $n = 0$  to  $95$ ) bits and the  $IELSR_n.DTCE$  ( $n = 0$  to  $95$ ) bit to 1

- If the IRQ pin is to be used for DMAC activation, set the DMAC. DELSRn.DELS[8:0] ( n = 0 to 7) bits.

## 13.6 Non-Maskable Interrupt Operation

The following sources can trigger a non-maskable interrupt:

- NMI pin interrupt
- Oscillation stop detection interrupt
- WDT underflow/refresh error interrupt
- IWDT underflow/refresh error interrupt
- Voltage monitor 1 interrupt
- Voltage monitor 2 interrupt
- Common memory error interrupt
- Bus error interrupt
- CPU Lockup error interrupt

Non-maskable interrupts can only be used with the CPU, not to activate the DTC. Non-maskable interrupts take precedence over all other interrupts. The non-maskable interrupt states can be verified in the Non-Maskable Interrupt Status Register (NMISR). Confirm that all bits in the NMISR are 0 before returning from the NMI handler.

There may be a difference in processing speed between the CPU and ICU, and the CPU may exit the interrupt handler before clearing the NMISR. Then the CPU will accidentally jump to the NMI handler again. To avoid this, be sure to read NMISR before exiting the NMI handler and make sure that NMISR is cleared before exiting the NMI handler.

Non-maskable interrupts are disabled by default. To use non-maskable interrupts:

1. Clear the NMICR.NFLTEN bit to 0 (digital filter disabled).
2. Set the NMIMD bit, NFCLKSEL[1:0] bits, and NFLTEN bit of NMICR register.
3. Write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
4. Enable the non-maskable interrupt by writing 1 to the associated bit in the Non-Maskable Interrupt Enable Register (NMIER).

After 1 is written to the NMIER register, subsequent write access to the NMIEN bit in NMIER is ignored. An NMI cannot be disabled when enabled, except by a reset.

### 13.6.1 Correspondence to TrustZone-M by NMI

The NMI security is set by AIRCR.BFHFNMIN.

Although there is only one NMI per CPU, multiple factors can be set. This section describes the procedure for mixing Secure and NonSecure programs of NMI. When doing so, the NMI-related registers of the ICU are set to Secure.

Register related to Non-Maskable Interrupt :

- NMISR
- NMIER
- NMICLR
- NMICR

Figure 13.4 shows the flow.

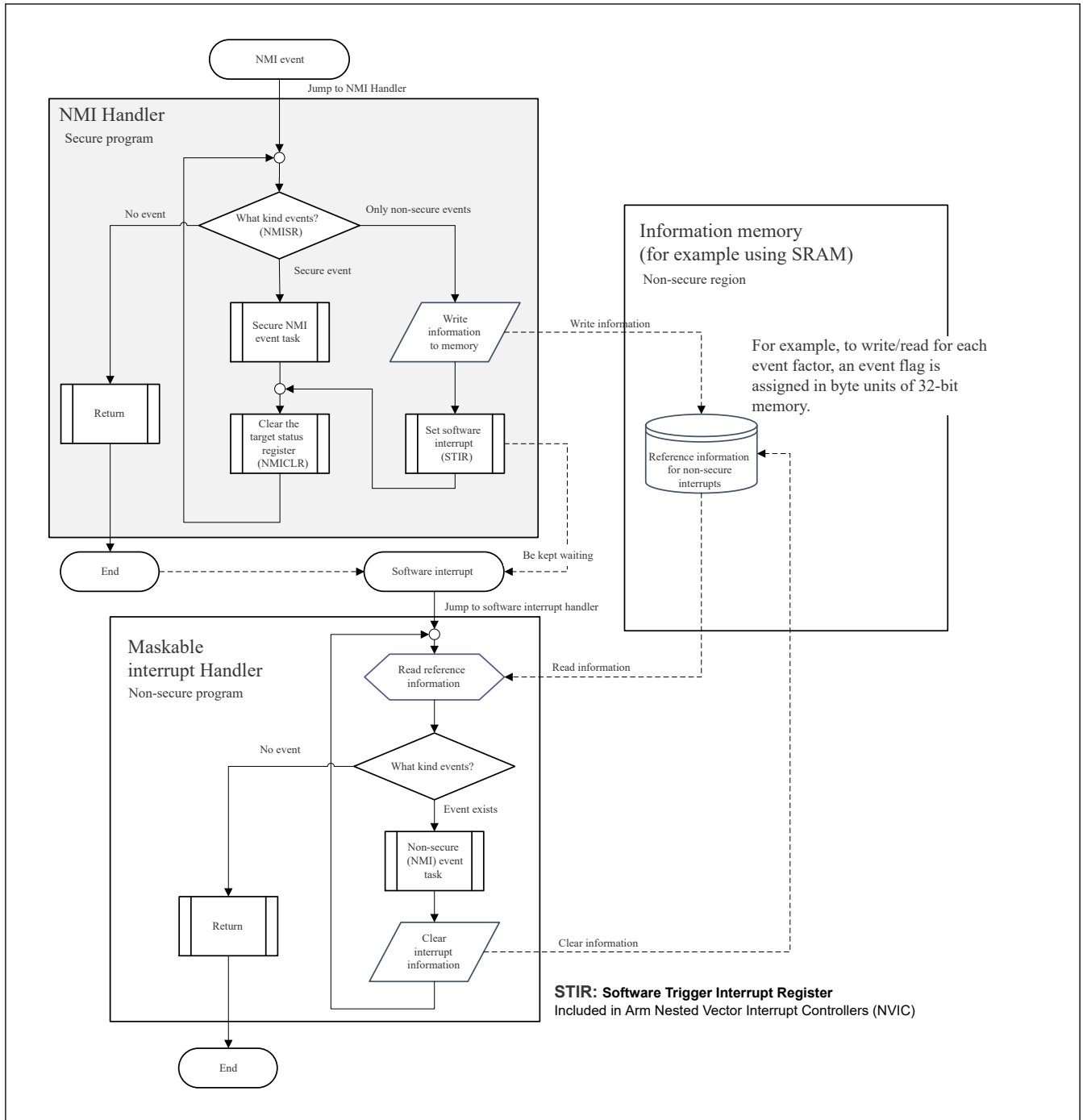


Figure 13.4 Correspondence to TrustZone-M by NMI

## 13.7 Security

### 13.7.1 Security Related to CPU Interrupt Inputs

#### Non-maskable interrupt

Arm CPU NMI Secure is modified by AIRCR.BFHFNMINs and is managed by the software developer who manages Secure.

The following registers must follow the secure attribution set in AIRCR.BFHFNMINs.

- ICU.NMISR
- ICU.NMIER

- ICU.NMICLR

Therefore, the secure attribute set by ICUSARB must match the secure attribute set by AIRCR.BFHFNMINS.

### Maskable interrupts

The secure of maskable interrupts are set in the Arm CPU NVIC internal registers (NVIC\_ITNS0 to NVIC\_ITNS15).

Which maskable interrupts are secured is controlled by the software developer who manages secure.

The following registers must follow the secure attribution set in NVIC\_ITNS0 to NVIC\_ITNS15.

- ICU.IELSRn (n = 0 to 95)

Therefore, the secure attribute set by ICUSARG, ICUSARH, and ICUSARI must match the secure attribute set by NVIC\_ITNS0 to NVIC\_ITNS15.

## 13.7.2 Trusted Interrupt Management

The secure-interrupt should not be visible to non-secure-operation. Because there is a risk that vital information will be leaked to the attacker, the on-chip interrupt network can route any interrupt to secure or non-secure world. However, secure interrupt routing must be suppressed so that it can only be configured from a secure world.

When set TEVTRCR.TEVTE = 1, secure program manages the selection of interrupt factors. It protects non-secure programs from using and monitoring secure interrupt factors without permission. Specifically, when TEVTRCR.TEVTE = 1, the write permission of IELSR.IELS [8:0] is limited to the secure attribute, and the write by the non-secure attribute is ignored. TEVTRCR.TEVTE protects only IELS [8:0], and the security attributes of IR and DTCE follow the settings of ICUSARG, ICUSARH, and ICUSARI. If TEVTRCR.TEVTE = 1, it is necessary to set all IELS including non-secure interrupts in the secure program or prepare a secure API to set IELS in response to the request from the non-secure program.

## 13.7.3 Trusted IELSR Setting Procedure

### 13.7.3.1 Case where Secure program sets all IELSRs

Case where all settings are performed in the secure initial sequence.

#### Initial IELSR setting procedure after system reset release

1. Secure program sets TEVTE=1
2. Secure program sets security attributes of all interrupts (NVIC\_ITNS, Security Attribution of ICU.IELSRn.IELS)
3. Secure program selects all interrupt source (ICU.IELSRn.IELS)
4. Secure program allows secure interrupts (NVIC\_ISER for secure interrupt)
5. Secure program jumps to non-secure program
6. Non-secure program allows non-secure interrupts (NVIC\_ISER for non-secure interrupt)

### 13.7.3.2 Case where Non-secure program sets IELSRn by Secure API

Case of setting upon receiving a request from a non-secure program.

#### Initial IELSR setting procedure after system reset release

1. Secure program sets TEVTE=1
2. Secure program sets security attributes of all interrupts (NVIC\_ITNS, Security Attribution of ICU.IELSRn.IELS)
3. Secure program selects secure interrupt source (ICU.IELSRn.IELS)
4. Secure program enables secure interrupt (NVIC\_ISER for secure interrupt)
5. Secure program jumps to non-secure program
6. Non-secure program calls secure program (API call)
7. Secure program selects the cause of non-secure interrupt (ICU.IELSRn.IELS)
8. Secure program returns to non-secure program
9. Non-secure program allows non-secure interrupt (NVIC\_ISER for non-secure interrupt)

### 13.7.3.3 IELSR release procedure when TEVTE=1

#### < When clearing interrupt settings of secure attribute by the secure program >

1. Clear the interrupt source setting (IELSRn.IELS = 0x00)
2. Clear the interrupt status flag (ICU.IELSRn.IR = 0)
3. Clear the Interrupt Clear Enable register (NVIC\_ICPR)

#### <When clearing interrupt settings of non-secure attribute by the secure program >

1. Clear the interrupt source setting (IELSRn.IELS = 0x00)
2. Clear the interrupt status flag by using the non-secure alias address (IELSRn.IR = 0)
3. Clear the Interrupt Clear Enable register (NVIC\_ICPR)

Secure can also rewrite NVIC\_ICPRn\_NS .

#### < When clearing interrupt settings of secure attribute by the non-secure program >

It cannot be canceled.

#### < When clearing interrupt settings of non-secure attribute by non-secure program >

1. Non-secure program calls secure program (API call)
2. Clear the interrupt source setting (IELSRn.IELS = 0x00)
3. Return from secure program to non-secure program
4. Clear the interrupt status flag (IELSRn.IR = 0)
5. Clear the Interrupt Clear Enable register (NVIC\_ICPR)

## 13.8 Return from Low Power Modes

[Table 13.4](#) lists the interrupt sources that can be used to exit Sleep, Deep Sleep, Software Standby mode or Deep Software Standby mode. For more information, see [section 10, Low Power Modes](#).

Note: The return factor from low power mode must have a priority that the CPU can accept.

### 13.8.1 Return from CPU Sleep Mode

To return from CPU Sleep mode in response to an interrupt:

#### non-maskable interrupt

- Use the NMIER register to enable the target interrupt request.

#### maskable interrupt

- Select the CPU as the interrupt request destination.
- Enable the interrupt in the NVIC.

### 13.8.2 Return from CPU Deep Sleep Mode

The ICU returns from CPU Deep Sleep mode using a non-maskable interrupt or a maskable interrupt. For maskable interrupt of canceling source, see [Table 13.4](#).

To return from CPU Deep Sleep mode:

#### Non-maskable interrupt

Use the NMIER register to enable the target interrupt request.

#### Maskable interrupt

1. Select the interrupt source that enables return from CPU Deep Sleep mode.

2. Use the WUPEN register to enable the target interrupt request.
3. Select the CPU as the interrupt request destination.
4. Enable the interrupt in the NVIC.

Interrupt requests through the IRQ<sub>i</sub> (i = 0 to 15) pins that do not satisfy these conditions are not detected while the clock is stopped in CPU Deep Sleep mode. Similarly, it cannot detect a request for a non-maskable interrupt from a request source whose clock is stopped in CPU Deep Sleep mode. Therefore, those events cannot be used as return factors. For example, the CPU cannot return from software standby mode because the following NMI events do not occur.

- CPU Lockup error interrupt

### 13.8.3 Return from Software Standby Mode

MCU goes to the Software Standby mode via CPU Deep sleep state.

The ICU returns from Software Standby mode using a non-maskable interrupt or a maskable interrupt. For maskable interrupt of canceling source, see [Table 13.4](#).

Waking up from Software Standby mode is triggered by the same event as CPU Deep Sleep mode. No event occurs from the function that stops in Software Standby mode. Therefore, those events cannot be used as return factors. For example, the CPU cannot return from Software Standby mode because the following NMI events do not occur.

- WDT underflow/refresh error
- Common memory error interrupt
- Bus error interrupt
- CPU Lockup error interrupt

### 13.9 Using the WFI Instruction with Non-Maskable Interrupts

Whenever a WFI instruction is executed, confirm that all status flags in the NMISR register are 0.

### 13.10 Reference

- Arm Limited., Arm<sup>®</sup> Cortex<sup>®</sup>-M85 Processor Technical Reference Manual (101924\_0002\_05\_en)



## 14. Buses

### 14.1 Overview

Table 14.1 lists the bus masters and bus slaves, Figure 14.1 shows the system bus configuration, and Table 14.3 lists the addresses assigned for each bus.

**Table 14.1 Bus specifications**

Classification	Bus master/ slave name	Sync clock	Description
Bus masters	CPUMAXIBI (Arm® Cortex®-M85)	CPUCLK	Connected to the CPU Master-AXI (M-AXI) Interface
	CPUPAHBI (Arm® Cortex®-M85)	CPUCLK	Connected to the CPU Peripheral AHB (P-AHB) Interface
	DMAC/DTCBI	ICLK	Connected to the DMAC/DTC Interface
	EDMACBI	PCLKA	Connected to the Ether DMAC Interface
	CEUBI	PCLKA	Connected to the CEU Interface
Bus slaves	FHBI	ICLK	Connected to Code Flash memory and Configuration area
	FLBI	FCLK	Connected to Data Flash memory and FACI
	CPUSAHBI	CPUCLK	Connected to CPU S-AHB
	S1BI	ICLK	Connected to SRAM1
	STBYSBI	ICLK	Connected to Standby SRAM
	EOBI	PCLKA	Connected to the OSPI (External Memory Interface)
	PBBI	PCLKB	Connected to peripheral modules synchronizes with PCLKB
	PABI	PCLKA	Connected to peripheral modules synchronizes with PCLKA
	PIBI	ICLK	Connected to peripheral modules synchronizes with ICLK
	ICUBI	ICLK	Connected to ICU controller
	PSBI	ICLK	Connected to peripheral system modules (MPU, SRAM, Debug component, Flash controller, Bus controller, common ICU controller, DMAC/DTC, CPU controller and security attribution controller)
		PCLKB	Connected to peripheral system module. (system controller)
	DCLK	Connected to peripheral system module. (debug controller)	

Note:

- CPUMAXIBI : CPU M-AXI Bus Interface
- CPUPAHBI : CPU P-AHB Bus Interface
- DMAC/DTCBI : DMAC/DTC Bus Interface
- EDMACBI : EDMAC (Ether) Bus Interface
- CEUBI : CEU Bus Interface
- FHBI : Flash High-speed Bus Interface
- FLBI : Flash Low-speed Bus Interface
- CPUSAHBI : CPU S-AHB Interface
- S1BI : SRAM1 Bus Interface
- STBYSBI : Standby SRAM Bus Interface
- EOBI : External memory OSPI Bus Interface
- PBBI : Peripheral synchronizes with PCLKB Bus Interface
- PABI : Peripheral synchronizes with PCLKA Bus Interface
- PIBI : Peripheral synchronizes with ICLK Bus Interface
- ICUBI : ICU Controller Bus Interface
- PSBI : Peripheral System Bus Interface

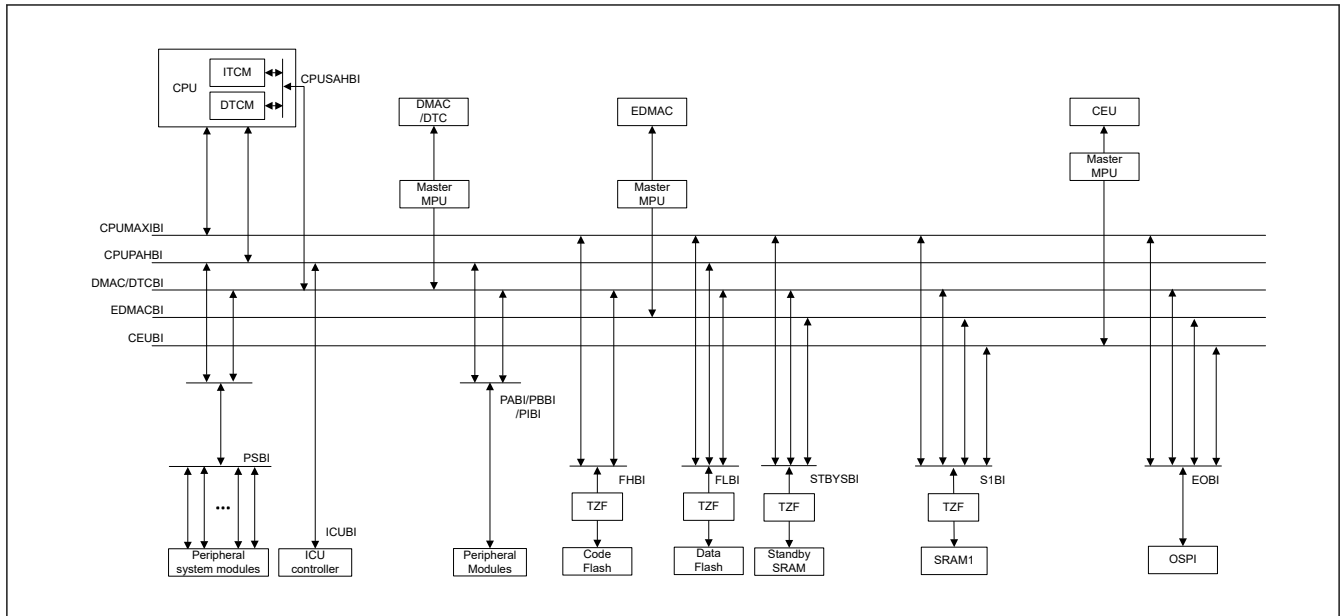


Figure 14.1 System bus connection

Table 14.2 System bus access path

		Master				
Name		CPUMAXIBI	CPUPAHBI	DMAC/DTCBI	EDMACBI	CEUBI
Slave	PSBI	F	T	T	F	F
	ICUBI	F	T	F	F	F
	CPUSAHBI	F	F	T	F	F
	PIBI	F	T	T	F	F
	PABI	F	T	T	F	F
	PBBI	F	T	T	F	F
	FHBI	T	F	T	F	F
	FLBI	T	T	T	F	F
	STBYSBI	T	F	T	T	F
	S1BI	T	F	T	T	T
	EOBI	T	F	T	T	T

Note: TZF is TrustZone Filter for memory resources. CPUMAXIBI accesses Data Flash Memory and Configuration area through FLBI. CPUPAHBI accesses FACI (Peripheral region) through FLBI.

T : The master can access slaves

F : The master cannot access slaves

Table 14.3 Addresses assigned for each slave (1 of 2)

Region	Address	Bus Interface	Area
Code	0x0000_0000 to 0x0000_FFFF (Non-secure callable or Secure) 0x1000_0000 to 0x1000_FFFF (Non-secure)	CPUSAHBI	ITCM
	0x0200_0000 to 0x0227_FFFF (Non-secure callable or Secure) 0x1200_0000 to 0x1227_FFFF (Non-secure)	FHBI	Code Flash memory
	0x0300_8000 to 0x0300_A2FF (Non-secure callable or Secure) 0x1300_8000 to 0x1300_A2FF (Non-secure)	FHBI	Factory Flash and option-setting memory

**Table 14.3** Addresses assigned for each slave (2 of 2)

Region	Address	Bus Interface	Area
Data	0x2000_0000 to 0x2000_FFFF (Non-secure callable or Secure) 0x3000_0000 to 0x3000_FFFF (Non-secure)	CPUSAHBI	DTCM
	0x2206_0000 to 0x220D_FFFF (Non-secure callable or Secure) 0x3206_0000 to 0x320D_FFFF (Non-secure)	S1BI	SRAM1
	0x2600_0000 to 0x2600_03FF (Non-secure callable or Secure) 0x3600_0000 to 0x3600_03FF (Non-secure)	STBYSBI	Standby SRAM
	0x2700_0000 to 0x2700_2FFF (Non-secure callable or Secure) 0x3700_0000 to 0x3700_2FFF (Non-secure)	FLBI	Data Flash memory
	0x2703_0050 to 0x2703_03FF (Non-secure callable or Secure) 0x3703_0050 to 0x3703_03FF (Non-secure)	FLBI	option-setting memory
Peripheral	0x4000_0000 to 0x4000_0FFF (Secure) 0x5000_0000 to 0x5000_0FFF (Non-secure)	PSBI	MPU controller
	0x4000_2000 to 0x4000_2FFF (Secure) 0x5000_2000 to 0x5000_2FFF (Non-secure)		SRAM controller
	0x4000_4000 to 0x4000_4FFF (Secure) 0x5000_4000 to 0x5000_4FFF (Non-secure)		BUS controller
	0x4000_6000 to 0x4000_6FFF (Secure) 0x5000_6000 to 0x5000_6FFF (Non-secure)		Common ICU controller
	0x4000_8000 to 0x4000_8FFF (Secure) 0x5000_8000 to 0x5000_8FFF (Non-secure)		Security attribution controller
	0x4000_A000 to 0x4000_AFFF (Secure) 0x5000_A000 to 0x5000_AFFF (Non-secure)		DMAC/DTC
	0x4000_C000 to 0x4000_CFFF (Secure) 0x5000_C000 to 0x5000_CFFF (Non-secure)	ICUBI	ICU controller
	0x4000_F000 to 0x4000_FFFF (Secure) 0x5000_F000 to 0x5000_FFFF (Non-secure)	PSBI	CPU controller
	0x4001_0000 to 0x4001_AFFF (Secure) 0x5001_0000 to 0x5001_AFFF (Non-secure)		Debug component
	0x4001_B000 to 0x4001_BFFF (Secure) 0x5001_B000 to 0x5001_BFFF (Non-secure)		Debug controller
	0x4001_C000 to 0x4001_CFFF (Secure) 0x5001_C000 to 0x5001_CFFF (Non-secure)		Flash controller
	0x4001_E000 to 0x4001_EFFF (Secure) 0x5001_E000 to 0x5001_EFFF (Non-secure)		System controller
	0x4010_0000 to 0x401F_FFFF (Secure) 0x5010_0000 to 0x501F_FFFF (Non-secure)		FLBI
	0x4020_0000 to 0x402F_FFFF (Secure) 0x5020_0000 to 0x502F_FFFF (Non-secure)	PBBI	Peripheral synchronizes with PCLKB
	0x4030_0000 to 0x403F_FFFF (Secure) 0x5030_0000 to 0x503F_FFFF (Non-secure)	PABI	Peripheral synchronizes with PCLKA
0x4040_0000 to 0x404F_FFFF (Secure) 0x5040_0000 to 0x504F_FFFF (Non-secure)	PIBI	Peripheral synchronizes with ICLK	
External RAM	0x8000_0000 to 0x9FFF_FFFF	EOBI	OSPI

## 14.2 Description of Buses

### 14.2.1 Arbitration

For arbitration between masters in each slave, fixed-priority and round-robin methods can be selected. For details, see [section 14.3.4. BUSSABT0<slave> : Bus Slave Arbitration Control Register 0\(<slave> = FLBI, STBYSBI, EOBI, PBBI, PABI, PIBI, PSBI\)](#) and [section 14.3.5. BUSSABT1<slave> : Bus Slave Arbitration Control Register 1\(<slave> = FHBI, S1BI\)](#).

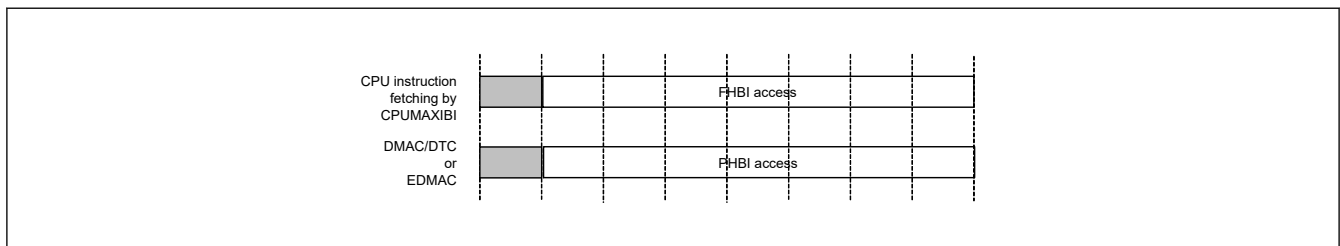
### 14.2.2 External Bus

The bus system provides an external space for the OSPI (EOBI). See [section 35, Octal Serial Peripheral Interface \(OSPI\)](#).

### 14.2.3 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave interface. For example, if the CPU is fetching an instruction from Code Flash and an operand from SRAM1, the DMAC can handle transfers between a peripheral module and the external memory interface at the same time.

[section 14.2.3. Parallel Operation](#) shows an example of parallel operations. In this example, the CPU uses CPUMAXIBI to access FHBI. Furthermore, the DMAC/DTC or EDMAC simultaneously accesses the peripheral bus during access to FHBI and S1BI by the CPU.



**Figure 14.2 Example of parallel operations**

### 14.2.4 Restrictions

#### 14.2.4.1 Endianness constraint

Memory space must be little-endian to execute code of the Arm<sup>®</sup> Cortex<sup>®</sup>-M85 core.

#### 14.2.4.2 Bufferable write access

If an access violation occurs due to bufferable write access, an error response is not returned to the bus master, but an NMI or RESET request is issued depending on the setting of the BUSOAD register. For more information, see [section 14.4.2. Operations When a Bus Error Occurs](#).

#### 14.2.4.3 Access to the reserved area of FLBI

Access to the reserved area of FLBI is prohibited. Operation is not guaranteed if accessed.

## 14.3 Register Descriptions

### 14.3.1 BUSSARA : Bus Security Attribution Register A

Base address: CPSCU = 0x4000\_8000  
 CPSCU\_NS = 0x5000\_8000

Offset address: 0x100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSSA0	Bus Security Attribution A0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

### BUSSA0 bit (Bus Security Attribution A0)

Security attributes of registers for Slave Bus Control Register. The target registers are as follows:

- BUSMABT
- BUSSABT0<slave>  
<slave> = FLBI/STBYSBI/EOBI/PBBI/PABI/PIBI/PSBI
- BUSSABT1<slave>  
<slave> = FHBI/S1BI
- BUSDIVBYP

### 14.3.2 BUSSARB : Bus Security Attribution Register B

Base address: CPSCU = 0x4000\_8000  
CPSCU\_NS = 0x5000\_8000

Offset address: 0x104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSS B0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSSB0	Bus Security Attribution B0 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

### BUSSB0 bit (Bus Security Attribution B0)

Security attributes of bus error related registers. The target registers are as follows:

- BUSnERRCLR
- MBWERRCLR
- SBWERRCLR
- BUSOAD
- BUSOADPT

### 14.3.3 BUSDIVBYP : Bus Divider Bypass Register

Base address: BUS = 0x4000\_3000  
BUS\_NS = 0x5000\_3000

Offset address: 0x1300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPU0 SBPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EDMA BPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EDMABPE	Divider for EDMACBI bypass enable 0: Disable 1: Enable	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
16	CPU0SBPE	Divider for CPUSAHBI bypass enable 0: Disable 1: Enable	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

#### EDMABPE bit (Divider for EDMACBI bypass enable)

This bit enables the through transfer between master and slave by bypassing the frequency divider implemented for EDMACBI. It can reduce the waiting time of 1 PCLKA cycle. This bit can be set to 1 only when ICLK frequency is the same as PCLKA frequency. This bit cannot be switched during EDMACBI operation.

#### CPU0SBPE bit (Divider for CPUSAHBI bypass enable)

This bit enables the through transfer between master and slave by bypassing the frequency divider implemented for CPUSAHBI. It can reduce the waiting time of 1 ICLK cycle. This bit can be set to 1 only when CPUCLK frequency is the same as ICLK frequency. This bit cannot be switched during CPUSAHBI operation.

### 14.3.4 BUSSABT0<slave> : Bus Slave Arbitration Control Register 0(<slave> = FLBI, STBYSBI, EOBI, PBBI, PABI, PIBI, PSBI)

Base address: BUS = 0x4000\_3000  
 BUS\_NS = 0x5000\_3000

Offset address: 0x1210 (BUSSABT0FLBI)  
 0x1248 (BUSSABT0STBYSBI)  
 0x1258 (BUSSABT0EOBI)  
 0x1260 (BUSSABT0PBBI)  
 0x1268 (BUSSABT0PABI)  
 0x1270 (BUSSABT0PIBI)  
 0x1278 (BUSSABT0PSBI)

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

Bit	Symbol	Function	R/W
0	ARBS	Arbitration Select for <slave> 0: Fixed priority 1: Round-robin	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

<slave> is bus interface unit name for slave.

This register specifies the arbitration method for <slave> buses.

(<slave> =FLBI/STBYSBI/PBBI/PABI/PIBI/PSBI)

#### ARBS bit (Arbitration Select for <slave>)

The ARBS bit specifies the priority between bus masters connected to <slave>. For fixed priority, see [Table 14.4](#). For round-robin, see [Table 14.5](#).

**Table 14.4 Bus priorities with fixed-priority arbitration (ARBS = 0)**

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT0FLBI	FLBI	DMAC/DTC > CPU M-AXI > CPU P-AHB
BUSSABT0STBYSBI	STBYSBI	EDMAC > DMAC/DTC > CPU M-AXI
BUSSABT0PBBI	PBBI	DMAC/DTC > CPU P-AHB
BUSSABT0PABI	PABI	DMAC/DTC > CPU P-AHB
BUSSABT0PIBI	PIBI	DMAC/DTC > CPU P-AHB
BUSSABT0PSBI	PSBI	DMAC/DTC > CPU P-AHB

Note 1. ">" represents left side has high priority.

**Table 14.5 Bus priorities with round-robin arbitration (ARBS = 1) (1 of 2)**

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT0FLBI	FLBI	DMAC/DTC <-> CPU M-AXI <-> CPU P-AHB
BUSSABT0STBYSBI	STBYSBI	EDMAC <-> DMAC/DTC <-> CPU M-AXI
BUSSABT0PBBI	PBBI	DMAC/DTC <-> CPU P-AHB
BUSSABT0PABI	PABI	DMAC/DTC <-> CPU P-AHB

**Table 14.5 Bus priorities with round-robin arbitration (ARBS = 1) (2 of 2)**

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT0PIBI	PIBI	DMAC/DTC <-> CPU P-AHB
BUSSABT0PSBI	PSBI	DMAC/DTC <-> CPU P-AHB

Note 1. “<->” represents round-robin.

### 14.3.5 BUSSABT1<slave> : Bus Slave Arbitration Control Register 1(<slave> = FHBI, S1BI)

Base address: BUS = 0x4000\_3000  
 BUS\_NS = 0x5000\_3000

Offset address: 0x1200 (BUSSABT1FHBI)  
 0x1220 (BUSSABT1S1BI)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ARBS[1:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ARBS[1:0]	Arbitration Select for <slave> 0 0: Fixed priority 0 1: Setting prohibited 1 0: Combination of round-robin and fixed priority 1 1: Round-robin	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

<slave> is bus interface unit name for slave.

This register specifies the arbitration method for <slave> buses.

(<slave> = FHBI/S1BI)

#### ARBS[1:0] bits (Arbitration Select for <slave>)

The ARBS bits specify the priority between bus masters connected to <slave>. For priority of each setting, see [Table 14.6](#), [Table 14.7](#) and [Table 14.8](#).

**Table 14.6 Bus priorities with fixed-priority arbitration (ARBS[1:0] = 00)**

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT1FHBI	FHBI	DMAC/DTC > CPU M-AXI
BUSSABT1S1BI	S1BI	CEU > EDMAC > DMAC/DTC > CPU M-AXI

Note 1. “>” represents left side has high priority.

**Table 14.7 Bus priorities with combination of round-robin and fixed priority arbitration (ARBS[1:0] = 10)**

Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT1FHBI	FHBI	DMAC/DTC > CPU M-AXI
BUSSABT1S1BI	S1BI	CEU > EDMAC > DMAC/DTC > CPU M-AXI

Note 1. “<->” represents round-robin. “>” represents left side has high priority.



**Table 14.8 Bus priorities with round-robin arbitration (ARBS[1:0] = 11)**

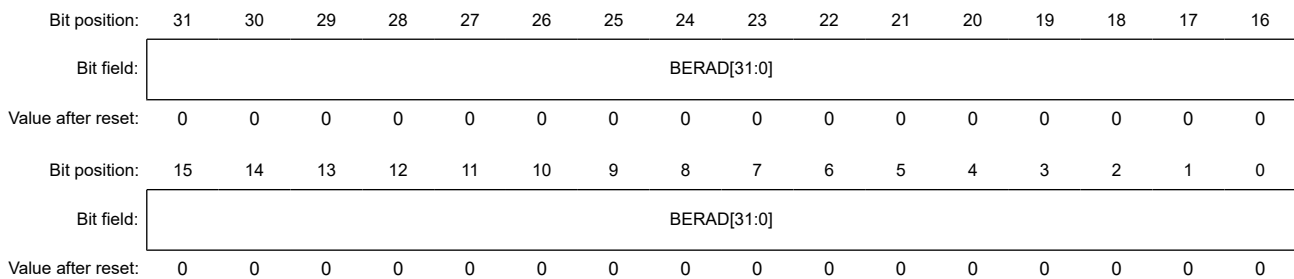
Bus Slave Arbitration Control Register	Slave interface	Priority order*1
BUSSABT1FHBI	FHBI	DMAC/DTC <-> CPU M-AXI
BUSSABT1S1BI	S1BI	CEU <-> EDMAC <-> DMAC/DTC <-> CPU M-AXI

Note 1. “<->” represents round-robin.

### 14.3.6 BUSnERRADD : BUS Error Address Register (n = 4, 5)

Base address: BUS = 0x4000\_3000  
 BUS\_NS = 0x5000\_3000

Offset address: 0x1800 + 0x10 × (n-1)



Bit	Symbol	Function	R/W
31:0	BERAD[31:0]	Bus Error Address When a bus error occurs, it stores an error address.	R

Note: S-TYPE-5, P-TYPE-2

Table 14.9 shows the register numbers associated with each bus type.

This register is cleared by set the BUSnERRCLR.ILERRCLR, BUSnERRCLR.MMERRCLR, or BUSnERRCLR.SLERRCLR bits or reset other than Bus error reset and Common memory error reset.

#### BERAD[31:0] bits (Bus Error Address)

The BERAD[31:0] bits indicates the address, when an error occurs on the associated bus. For detail of error that occurs by bus, see section 14.3.10. BUSnERRSTAT : BUS Error Status Register (n = 1 to 5, 10) and section 14.4. Bus Error Monitoring Section.

When an error occurs on the bus, the corresponding bit of ILERRSTAT, MMERRSTAT, SLERRSTAT in BUSnERRSTAT (n = 1 to 5, 10) is set to 1, at the same time, the BERAD[31:0] bits stores the address of the bus error access.

When the error address is set once, it is not updated until the value is cleared by the BUSnERRCLR Register.

When all of ILERRSTAT, MMERRSTAT, SLERRSTAT, MSERRSTAT in BUSnERRSTAT (n = 1 to 5, 10) are set to 0, the BERAD[31:0] bits are indefinite.

**Table 14.9 Associations between bus types and registers**

Bus type	Bus error status register	Bus error clear register	Bus error address register	Bus error RW register
CPUMAXIBI_R channel	BUS1ERRSTAT	BUS1ERRCLR	—	—
CPUMAXIBI_W channel	BUS2ERRSTAT	BUS2ERRCLR	—	—
CPUPAHBI	BUS3ERRSTAT	BUS3ERRCLR	—	—
DMAC/DTCBI	BUS4ERRSTAT	BUS4ERRCLR	BUS4ERRADD BMSA4ERRADD	BUS4ERRRW BMSA4ERRRW
EDMACBI	BUS5ERRSTAT	BUS5ERRCLR	BUS5ERRADD BMSA5ERRADD	BUS5ERRRW BMSA5ERRRW
CEUBI_W channel	BUS10ERRSTAT	BUS10ERRCLR	—	—

### 14.3.7 BUSnERRRW : BUS Error Read Write (n = 4, 5)

Base address: BUS = 0x4000\_3000  
 BUS\_NS = 0x5000\_3000

Offset address: 0x1804 + 0x10 × (n-1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RWST AT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RWSTAT	Error access Read/Write Status The status at the time of the error. 0: Read access 1: Write access	R
7:1	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2

This register is cleared by a reset other than the Bus error reset and Common memory error reset, and the BUSnERRCLR.ILERRCLR, BUSnERRCLR.MMERRCLR and BUSnERRCLR.SLERRCLR.

Table 14.9 shows the register numbers associated with each bus type.

#### RWSTAT bit (Error access Read/Write Status)

The RWSTAT bit indicates the access status, write access or read access, when an error occurs on the associated bus.

For detail of error that occurs by bus, see section 14.3.10. BUSnERRSTAT : BUS Error Status Register (n = 1 to 5, 10) and section 14.4. Bus Error Monitoring Section.

When an error occurs on the bus, the corresponding bit of ILERRSTAT, MMERRSTAT, SLERRSTAT, in BUSnERRSTAT (n = 1 to 5, 10) is set to 1, at the same time, the RWSTAT bits stores the read/write status of the bus error access.

When all of ILERRSTAT, MMERRSTAT, SLERRSTAT in BUSnERRSTAT (n = 1 to 5, 10) are set to 0, the RWSTAT bit is indefinite.

When the RWSTAT bit is set once, it is not updated until the value is cleared by the BUSnERRCLR Register.

### 14.3.8 BMSAnERRADD : Bus Master Security Attribution Unit Error Address (n = 4, 5)

Base address: BUS = 0x4000\_3000  
 BUS\_NS = 0x5000\_3000

Offset address: 0x1900 + 0x10 × (n-1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSERAD[31:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSERAD[31:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	MSERAD[31:0]	Bus Master Security Attribution Unit Error Address When a Master Security Attribution Unit error occurs, It stores an error address.	R

Note: S-TYPE-5, P-TYPE-2

Note: This register is cleared by set the BUSnERRCLR.MSERRCLR bit or reset other than Bus error reset and Common memory error reset.

Table 14.9 lists the registers associated with each bus type.

**MSERAD[31:0] bits (Bus Master Security Attribution Unit Error Address)**

The MSERAD[31:0] bits indicates the address, when an error occurs on the associated bus.

For detail of error that occurs by bus, see [section 14.3.10. BUSnERRSTAT : BUS Error Status Register \(n = 1 to 5, 10\)](#) and [section 14.4. Bus Error Monitoring Section](#).

When an error occurs on the bus, the corresponding bit of MSERRSTAT in BUSnERRSTAT (n = 1 to 5, 10) is set to 1, at the same time, the MSERAD[31:0] bits stores the address of the bus error access.

When the error address is set once, it is not updated until the value is cleared by the BUSnERRCLR (n = 1 to 5, 10).

When all of MSERRSTAT in BUSnERRSTAT (n = 1 to 5, 10) is set to 0, the MSERAD[31:0] bits are indefinite.

**14.3.9 BMSAnERRRW : BUS Master Security Attribution Unit Error Read Write (n = 4, 5)**

Base address: BUS = 0x4000\_3000  
 BUS\_NS = 0x5000\_3000

Offset address: 0x1904 + 0x10 × (n-1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MSAR WSTA T
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MSARWSTAT	Master Security Attribution Unit error access Read/Write Status The status at the time of the error. 0: Read access 1: Write access	R
7:1	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2

Note: This register is cleared by set the BUSnERRCLR.MSERRCLR bits or reset other than Bus error reset and Common memory error reset.

Table 14.9 shows the register numbers associated with each bus type.

**MSARWSTAT bit (Master Security Attribution Unit error access Read/Write Status)**

The MSARWSTAT bit indicates the access status, write access or read access, when an error occurs on the associated bus.

For detail of error that occurs by bus, see [section 14.3.10. BUSnERRSTAT : BUS Error Status Register \(n = 1 to 5, 10\)](#) and [section 14.4. Bus Error Monitoring Section](#).

When an error occurs on the bus, the corresponding bit of MSERRSTAT in BUSnERRSTAT (n = 1 to 5, 10) is set to 1, at the same time, the MSARWSTAT bits stores the read/write status of the bus error access.

When the MSARWSTAT bit is set once, it is not updated until the value is cleared by the BUSnERRCLR (n = 1 to 5, 10).

When all of MSERRSTAT in BUSnERRSTAT (n = 1 to 5, 10) are set to 0, the MSARWSTAT bit is indefinite.

**14.3.10 BUSnERRSTAT : BUS Error Status Register (n = 1 to 5, 10)**

Base address: BUS = 0x4000\_3000  
 BUS\_NS = 0x5000\_3000

Offset address: 0x1A00 + 0x10 × (n - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MSER RSTAT	ILERR STAT	MMER RSTAT	—	—	SLER RSTAT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SLERRSTAT	Slave Bus Error Status 0: No error occurred 1: Error occurred	R
2:1	—	These bits are read as 0.	R
3	MMERRSTAT	Master MPU Error Status 0: No error occurred 1: Error occurred	R
4	ILERRSTAT	Illegal Address Access Error Status 0: No error occurred 1: Error occurred	R
5	MSERRSTAT	Master Security Attribution Unit Error Status 0: No error occurred 1: Error occurred	R
7:6	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2

This register is cleared by reset other than Bus error reset and Common memory error reset.

Table 14.9 shows the register numbers associated with each bus type.

When the bus error occurred to plural at the same time, the STAT bit becomes effective only 1 by the following order of priority. The left side is high priority.

Master Security attribution unit > Master-MPU Error > Illegal access Error, Slave TrustZone filter, Slave bus Error  
Illegal access Error, Slave TrustZone filter Error and Slave bus Error don't occur at the same time.

After one of MSERRSTAT, ILERRSTAT, MMERRSTAT, and SLERRSTAT are set once, MSERRSTAT, ILERRSTAT, MMERRSTAT, and SLERRSTAT are not renewed.

#### SLERRSTAT bit (Slave Bus Error Status)

When slave error or TrustZone filter error occurs by bus, BUSnERRSTAT.SLERRSTAT (n = 1 to 5, 10) is set to 1.

There is clearing conditions.

- Set BUSnERRCLR.SLERRCLR to 1.

Slave error is an error that occurs on a slave such as Time-out.

The SLERRSTAT bit is not set when debugger violate.

For detail of slave error that occurs by bus, see [section 14.4. Bus Error Monitoring Section](#).

#### MMERRSTAT bit (Master MPU Error Status)

When master mpu error occurs by bus, BUSnERRSTAT.MMERRSTAT (n = 1 to 5, 10) is set to 1.

There is clearing conditions.

- Set BUSnERRCLR.MMERRCLR to 1.

For detail of master mpu error that occurs by bus, see [section 15, Memory Protection Unit \(MPU\)](#).

#### ILERRSTAT bit (Illegal Address Access Error Status)

When illegal address access error occurs by bus and write access to FHBI occurs, BUSnERRSTAT.ILERRSTAT (n = 1 to 5, 10) is set to 1.

There is clearing conditions.

- Set BUSnERRCLR.ILLERRCLR to 1.

The ILERRSTAT bit is not set when debugger violate.

For detail of illegal address access error that occurs by bus, see [section 14.4. Bus Error Monitoring Section](#).

**MSERRSTAT bit (Master Security Attribution Unit Error Status)**

When MSAU error occurs by bus, BUSnERRSTAT.MSERRSTAT (n = 1 to 5, 10) is set to 1.

There is clearing conditions.

- Set BUSnERRCLR.MSERRCLR to 1.

For detail of MSAU error that occurs by bus, see [section 14.4. Bus Error Monitoring Section](#).

**14.3.11 BUSnERRCLR : BUS Error Clear Register (n = 1 to 5, 10)**

Base address: BUS = 0x4000\_3000  
BUS\_NS = 0x5000\_3000

Offset address: 0x1A08 + 0x10 × (n - 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MSER RCLR	ILERR CLR	MMER RCLR	—	—	SLER RCLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SLERRCLR	Slave Bus Error Clear Writing 1 to this bit clears the BUSnERRSTAT.SLERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT.	R/W <sup>1</sup>
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	MMERRCLR	Master MPU Error Clear Writing 1 to this bit clears the BUSnERRSTAT.MMERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT.	R/W <sup>1</sup>
4	ILERRCLR	Illegal Address Access Error Clear Writing 1 to this bit clears the BUSnERRSTAT.ILERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT.	R/W <sup>1</sup>
5	MSERRCLR	Master Security Attribution Unit Error Clear Writing 1 to this bit clears the BUSnERRSTAT.MSERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT.	R/W <sup>1</sup>
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note 1. Only 1 can be written to this bit. This bit is read as 0. Writing 0 to this bit has no effect.

[Table 14.9](#) shows the register numbers associated with each bus type.

When writing 1 to BUSnERRCLR (n = 1 to 5, 10), stop the bus access that causes an error in the corresponding bus master.

**SLERRCLR bit (Slave Bus Error Clear)**

Writing 1 to the SLERRCLR bit clears the BUSnERRSTAT.SLERRSTAT, BMSAnERRADD.MSERAD and BMSAnERRRW.MSARWSTAT (n = 1 to 5, 10).

This bit is read as 0.

**MMERRCLR bit (Master MPU Error Clear)**

Writing 1 to the MMERRCLR bit clears the BUSnERRSTAT.MMERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT (n = 1 to 5, 10).

This bit is read as 0.

**ILERRCLR bit (Illegal Address Access Error Clear)**

Writing 1 to the ILERRCLR bit clears the BUSnERRSTAT.ILERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT (n = 1 to 5, 10).

This bit is read as 0.

**MSERRCLR bit (Master Security Attribution Unit Error Clear)**

Writing 1 to the MSERRCLR bit clears the BUSnERRSTAT.MSERRSTAT, BUSnERRADD.BERAD and BUSnERRRW.RWSTAT (n = 1 to 5, 10).

This bit is read as 0.

**14.3.12 BUSOAD : BUS Operation After Detection Register**

Base address: BUS = 0x4000\_3000  
BUS\_NS = 0x5000\_3000

Offset address: 0x1000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BWERROAD	SLERROAD	ILERROAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ILERROAD	Illegal address access error operation after detection CPU 0: Only error response is returned. 1: Reset Other master 0: NMI 1: Reset	R/W
1	SLERROAD	Slave bus error operation after detection CPU 0: Only error response is returned. 1: Reset Other master 0: NMI 1: Reset	R/W
2	BWERROAD	Bufferable write error operation after detection All bus master 0: NMI 1: Reset	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: When an error occurs in a speculative access issued by CPU with OAD bit is set to 1, a system reset request may occur. In that case, it can be avoided by using Arm-MPU. See [section 14.5. References](#)[3] for the detail.

**ILERROAD bit (Illegal address access error operation after detection)**

The OAD bit specify operation when the illegal address access error is detected. When OAD bit is 0, the operation differs depending on the bus master. If CPU causes access violation, only error response is returned and NMI is not generated. If master other than CPU causes access violation, error response is returned and NMI is generated.

When OAD bit is 1, reset request is generated.

**SLERROAD bit (Slave bus error operation after detection)**

The OAD bit specify operation when the slave bus error is detected. When OAD bit is 0, the operation differs depending on the bus master. If CPU causes access violation, only error response is returned and NMI is not generated. If master other than CPU causes access violation, error response is returned and NMI is generated.

When OAD bit is 1, reset request is generated.

**BWERROAD bit (Bufferable write error operation after detection)**

The OAD bit specify operation when the bufferable write error is detected.

When OAD bit is 0, error response is returned and NMI is generated.

When OAD bit is 1, reset request is generated.

### 14.3.13 BUSOADPT : BUS Operation After Detection Protect Register

Base address: BUS = 0x4000\_3000  
 BUS\_NS = 0x5000\_3000

Offset address: 0x1004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: BUSOAD register writing is possible. 1: BUSOAD register writing is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key code These bits enable or disable writes to the PROTECT bit.	W

Note: S-TYPE-3, P-TYPE-2

#### PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

BUSOADPT.PROTECT controls the following registers:

- BUSOAD

When the PROTECT bit is set simultaneously, write 0xA5 to the KEY[7:0] bits using half word access.

#### KEY[7:0] bits (Key code)

The KEY[7:0] bits enable or disable writes to the PROTECT bit. When writing to the PROTECT bit simultaneously, write 0xA5 to the KEY[7:0] bits. When other values are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

### 14.3.14 MBWERRSTAT : Master Bufferable Write Error Status Register

Base address: BUS = 0x4000\_3000  
 BUS\_NS = 0x5000\_3000

Offset address: 0x1B00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	MBWE RR21	—	—	—	—	MBWE RR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MBWE RR8	—	—	—	—	—	—	MBWE RR1	MBWE RR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBWERR0	Master Bufferable Write Error 0: No bufferable write error in Master #0 1: Bufferable write error occurs in Master #0	R
1	MBWERR1	Master Bufferable Write Error 0: No bufferable write error in Master #1 1: Bufferable write error occurs in Master #1	R
7:2	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
8	MBWERR8	Master Bufferable Write Error 0: No bufferable write error in Master #8 1: Bufferable write error occurs in Master #8	R
15:9	—	These bits are read as 0.	R
16	MBWERR16	Master Bufferable Write Error 0: No bufferable write error in Master #16 1: Bufferable write error occurs in Master #16	R
20:17	—	These bits are read as 0.	R
21	MBWERR21	Master Bufferable Write Error 0: No bufferable write error in Master #21 1: Bufferable write error occurs in Master #21	R
31:22	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2

Note: This register is cleared by reset other than Bus error reset and Common memory error reset.

### MBWERRn bit (Master Bufferable Write Error Clear n)

When bufferble write access error occurs by MBWERRSTAT. MBWERRn is set to 1.

There are clearing conditions.

- Set MBWERRCLR. MBWERRCn to 1

Table 14.10 lists the registers associated with each bus master type.

**Table 14.10 Associations between bus master types and bits**

Bus master type	Master bufferable write error status bit	Master bufferable write error clear bit
CPUMAXIBI_W channel	MBWERR0	MBWECLR0
CPUPAHBI	MBWERR1	MBWECLR1
DMAC/DTCBI	MBWERR8	MBWECLR8
EDMACBI	MBWERR16	MBWECLR16
CEUBI_W channel	MBWERR21	MBWECLR21

### 14.3.15 MBWERRCLR : Master Bufferable Write Error Clear Register

Base address: BUS = 0x4000\_3000  
BUS\_NS = 0x5000\_3000

Offset address: 0x1B08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	MBWE CLR21	—	—	—	—	MBWE CLR16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MBWE CLR8	—	—	—	—	—	—	MBWE CLR1	MBWE CLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBWECLR0	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR0 flag.	R/W <sup>1</sup>
1	MBWECLR1	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR1 flag.	R/W <sup>1</sup>
7:2	—	These bits are read as 0.	R/W <sup>1</sup>



Bit	Symbol	Function	R/W
8	MBWECLR8	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR8 flag.	R/W <sup>1</sup>
15:9	—	These bits are read as 0.	R/W <sup>1</sup>
16	MBWECLR16	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR16 flag.	R/W <sup>1</sup>
20:17	—	These bits are read as 0.	R/W <sup>1</sup>
21	MBWECLR21	Master Bufferable Write Error Clear Writing 1 to this bit clears the MBWERRSTAT.MBWERR21 flag.	R/W <sup>1</sup>
31:22	—	These bits are read as 0.	R/W <sup>1</sup>

Note: S-TYPE-3, P-TYPE-2

Note 1. Only 1 can be written to this bit.

### MBWECLRn bit (Master Bufferable Write Error Clear n)

Writing 1 to this bit clears the MBWERRSTAT.MBWERRn flag.

This bit is read as 0.

### 14.3.16 SBWERRSTAT : Slave Bufferable Write Error Status Register

Base address: BUS = 0x4000\_3000  
 BUS\_NS = 0x5000\_3000

Offset address: 0x1B20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SBWE RR12	SBWE RR11	SBWE RR10	SBWE RR9	SBWE RR8	SBWE RR7	—	SBWE RR5	SBWE RR4	—	SBWE RR2	SBWE RR1	SBWE RR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SBWERR0	Slave Bufferable Write Error 0: No bufferable write error in Slave #0 1: Bufferable write error occurs in Slave #0	R
1	SBWERR1	Slave Bufferable Write Error 0: No bufferable write error in Slave #1 1: Bufferable write error occurs in Slave #1	R
2	SBWERR2	Slave Bufferable Write Error 0: No bufferable write error in Slave #2 1: Bufferable write error occurs in Slave #2	R
3	—	This bit is read as 0.	R
4	SBWERR4	Slave Bufferable Write Error 0: No bufferable write error in Slave #4 1: Bufferable write error occurs in Slave #4	R
5	SBWERR5	Slave Bufferable Write Error 0: No bufferable write error in Slave #5 1: Bufferable write error occurs in Slave #5	R
6	—	This bit is read as 0.	R
7	SBWERR7	Slave Bufferable Write Error 0: No bufferable write error in Slave #7 1: Bufferable write error occurs in Slave #7	R

Bit	Symbol	Function	R/W
8	SBWERR8	Slave Bufferable Write Error 0: No bufferable write error in Slave #8 1: Bufferable write error occurs in Slave #8	R
9	SBWERR9	Slave Bufferable Write Error 0: No bufferable write error in Slave #9 1: Bufferable write error occurs in Slave #9	R
10	SBWERR10	Slave Bufferable Write Error 0: No bufferable write error in Slave #10 1: Bufferable write error occurs in Slave #10	R
11	SBWERR11	Slave Bufferable Write Error 0: No bufferable write error in Slave #11 1: Bufferable write error occurs in Slave #11	R
12	SBWERR12	Slave Bufferable Write Error 0: No bufferable write error in Slave #12 1: Bufferable write error occurs in Slave #12	R
31:13	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-2

Note: This register is cleared by reset other than Bus error reset and Common memory error reset.

### SBWERRn bit (Slave Bufferable Write Error n)

When bufferable write access error occurs by SBWERRSTAT.SBWERRn is set to 1.

There are clearing conditions.

- Set SBWERRCLR.SBWECLRn to 1.

Table 14.11 lists the registers associated with each bus slave type.

**Table 14.11 Associations between bus slave types and bits**

Bus slave type	Slave bufferable write error status bit	Slave bufferable write error clear bit
FHBI	SBWERR0	SBWECLR0
FLBI	SBWERR1	SBWECLR1
CPUSAHBI	SBWERR2	SBWECLR2
S1BI	SBWERR4	SBWECLR4
STBYSBI	SBWERR5	SBWECLR5
EOBI	SBWERR7	SBWECLR7
PBBI	SBWERR8	SBWECLR8
PABI	SBWERR9	SBWECLR9
PIBI	SBWERR10	SBWECLR10
ICUBI	SBWERR11	SBWECLR11
PSBI	SBWERR12	SBWECLR12

### 14.3.17 SBWERRCLR : Slave Bufferable Write Error Clear Register

Base address: BUS = 0x4000\_3000  
 BUS\_NS = 0x5000\_3000

Offset address: 0x1B28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SBWE CLR12	SBWE CLR11	SBWE CLR10	SBWE CLR9	SBWE CLR8	SBWE CLR7	—	SBWE CLR5	SBWE CLR4	—	SBWE CLR2	SBWE CLR1	SBWE CLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SBWECLR0	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR0 flag.	R/W <sup>1</sup>
1	SBWECLR1	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR1 flag.	R/W <sup>1</sup>
2	SBWECLR2	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR2 flag.	R/W <sup>1</sup>
3	—	This bit is read as 0.	R/W <sup>1</sup>
4	SBWECLR4	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR4 flag.	R/W <sup>1</sup>
5	SBWECLR5	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR5 flag.	R/W <sup>1</sup>
6	—	This bit is read as 0.	R/W <sup>1</sup>
7	SBWECLR7	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR7 flag.	R/W <sup>1</sup>
8	SBWECLR8	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR8 flag.	R/W <sup>1</sup>
9	SBWECLR9	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR9 flag.	R/W <sup>1</sup>
10	SBWECLR10	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR10 flag.	R/W <sup>1</sup>
11	SBWECLR11	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR11 flag.	R/W <sup>1</sup>
12	SBWECLR12	Slave Bufferable Write Error Clear Writing 1 to this bit clears the SBWERRSTAT.SBWERR12 flag.	R/W <sup>1</sup>
31:13	—	These bits are read as 0.	R/W <sup>1</sup>

Note: S-TYPE-3, P-TYPE-2

Note 1. Only 1 can be written to this bit.

#### SBWECLRn bit (Slave Bufferable Write Error Clear n)

Writing 1 to this bit clears the SBWERRSTAT.SBWERRn flag.

This bit is read as 0.

## 14.4 Bus Error Monitoring Section

This monitoring system monitors each individual area, and whenever it detects an error, it returns the error to the requesting master using the AHB-Lite error response protocol.

### 14.4.1 Bus Error Types

The following types of errors can occur on each bus:

- Master Security Attribution Unit error
- Bus master MPU error.
- Illegal address access.
- Slave TrustZone Filter error
- Bus error transmitted from each slave module.

Table 14.15 lists the address ranges where access leads to illegal address access errors. The reserved area in the slave does not trigger an illegal address access error. For more information on the bus master MPU, see [section 15, Memory Protection Unit \(MPU\)](#).

### 14.4.2 Operations When a Bus Error Occurs

In the following bufferable write access, the write response can be given from the midpoint in the following cases.

1. CPU M-AXI accesses FLBI, ROMBI and STBYSBI.
2. CPU P-AHB accesses an accessible area.
3. CPU P-AHB accesses illegal address.
4. DMA accesses EOBI, FLBI, PABI, PBBI, S1BI and PSBI.

If an access violation occurs due to these bufferable write accesses, an error response may not be returned to the bus master.

When a bufferable write access error that does not return an error response is detected, access is blocked and 1 is set in the bit corresponding to the error occurrence location of MBWERRSTAT or SBWERRSTAT. Furthermore, NMI request or Reset request is generated according to the setting of the BUSOAD.BWERROAD. Since MBWERRSTAT or SBWERRSTAT are held until reset other than Bus and Memory Error Reset or cleared by MBWERRCLR or SBWERRCLR, they can be confirmed in the BusFault Handler or Interrupt handler. Error address information and Error R/W information is not remain with a bufferable write access. If error information is required, please set the BUS Master to Non-bufferable access.

A bufferable write access error can occur in the following cases. In this case, BUSnERRSTAT is not set and NMI request or Reset request is not generated according to the setting of the BUSOAD.ILERRROAD or BUSOAD.SLERROAD.

1. CPU M-AXI accesses FLBI, ROMBI and STBYSBI and a slave error or slave TrustZone Filter error occurs.
2. CPU P-AHB accesses an accessible area and a slave error or slave TrustZone Filter error occurs.
3. CPU P-AHB accesses illegal address and a illegal address access error occurs.
4. DMA accesses EOBI, FLBI, PABI, PBBI, S1BI and PSBI and a slave error or slave TrustZone Filter error occurs.

**Table 14.12 Bus error notice method for Bufferable write access**

	DAP	CPU	DMAC/DTC	EDMAC	CEU
BUSOAD.BWERROAD = 0	BUSERR.NMI	BUSERR.NMI	BUSERR.NMI	BUSERR.NMI	BUSERR.NMI
BUSOAD.BWERROAD = 1	Reset	Reset	Reset	Reset	Reset

When a bus error occurs for an access other than the bufferable write, error access is blocked, and error operations determine by OAD setting and the bus master are performed. Table 14.13 shows OAD setting corresponding to the each error. Table 14.14 shows how to notify when a bus error occurs.

When OAD setting is NMI.

If CPU is the bus master, it only returns an error response to bus master. CPU generates BusFault when it receives error response. If master other than CPU is the bus master, error response is returned to the master and NMI is generated.

When OAD setting is Reset.

Error response is returned to the bus master and reset request is generated.

**Table 14.13 OAD setting corresponding to the each error**

Error type	OAD bit
MSAU Error	MSAOAD.OAD
MPU Error	MMPUOAD.OAD
Illegal access Error	BUSOAD.ILERRROAD
Slave Bus Error	BUSOAD.SLERROAD
Slave TrustZone Filter error	BUSOAD.SLERROAD

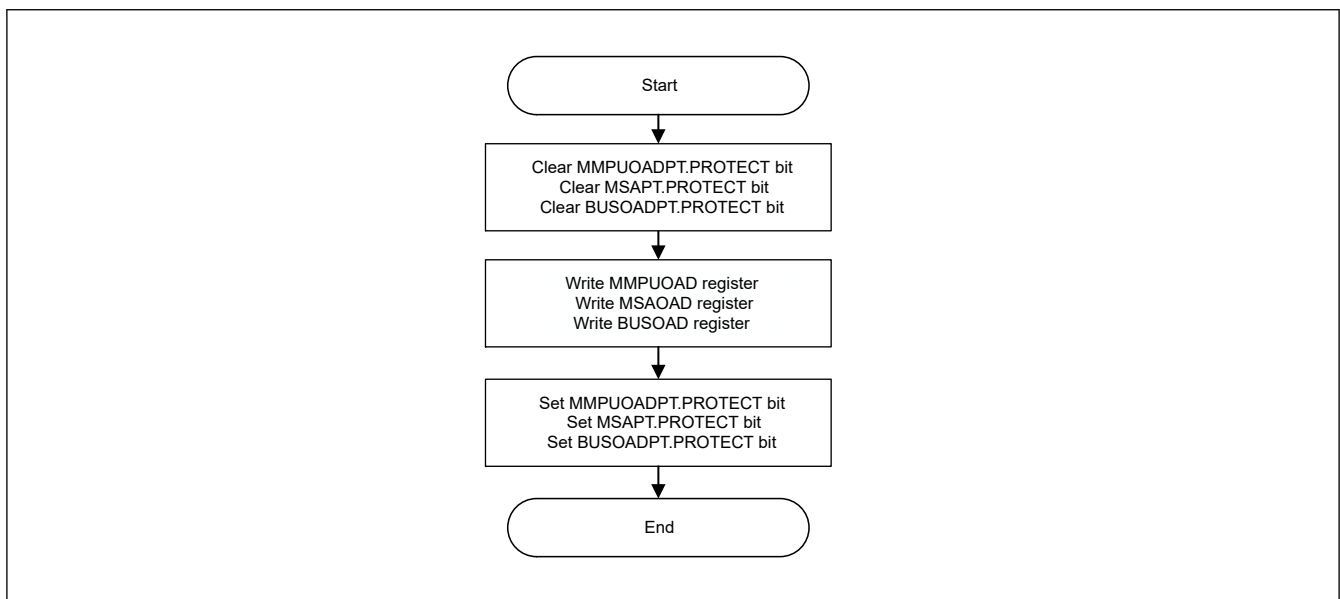
**Table 14.14 Bus error notice method for Non-bufferable write access**

OAD setting	DAP*2	CPU*1	DMAC/DTC	EDMAC	CEU
NMI	Error response only	CPU.BusFault	DMA0_TRANSERR*1 BUSERR.NMI	ETHER_EINT*1 BUSERR.NMI	BUSERR.NMI
Reset	Error response only	Reset	Reset	Reset	Reset

Note 1. DMAC/DTC and EDMAC can issue interrupt when it receives error response. This NMI is output to ICU and transmitted from ICU to CPU.

Note 2. Only when the bus master is DAP, only an error response is returned regardless of the BUSOAD.ILERRROAD or the BUSOAD.SLERROAD settings, and NMI and RESET requests are not generated.

Figure 14.3 shows the OAD setting flow after reset. During this register setting, stop the Bus Master.



**Figure 14.3 OAD setting flow after reset**

See for details of [section 15.3.1.23. MMPUOADPT : MPMU Operation After Detection Protect Register](#) and [section 15.3.1.22. MMPUOAD : MPMU Operation After Detection Register](#).

See for details of [section 40.8.15. MSAPT : Master Security Attribution Protect Register](#) and [section 40.8.14. MSAOAD : Master Security Attribution Operation After Detection Register](#).

**(1) Master Security Attribution Unit Error**

The bus master of other than CPU and DAP has MSAU to block access to Secure alias address from Non-secure masters. The CPU does not have MSAU since it has IDAU and SAU. When MSAU Error is detected, an Error response is returned to the Master. At the same time, store the address when MSAU Error occurs in BMSAnERRADD (n = 4, 5), store the RW information when MSAU Error occurs in BMSAnERRRW (n = 4, 5), set 1 to MSERRSTAT bit of BUSnERRSTAT (n = 1 to 5, 10). Furthermore, NMI request or Reset request is generated according to the setting of the MSAOAD.OAD. Since BMSAnERRADD (n = 4, 5), BMSAnERRRW (n = 4, 5), BUSnERRSTAT (n = 1 to 5, 10) are held until reset other than Bus error reset and Common memory error reset, or cleared by BUSnERRCLR (n = 1 to 5, 10), they can be

confirmed in the NMI handler or after reset. NMI request is generated only at the first MSAU Error after reset or clearing BUSnERRSTAT (n = 1 to 5, 10). MSERRSTAT bit by BUSnERRCLR (n = 1 to 5, 10).

## (2) Bus Master MPU Error

The bus master of other than CPU and DAP has Master-MPU for access control of the set address area. The CPU does not have Master-MPU since it has Arm MPU. When Master-MPU Error is detected, an Error response is returned to the Master. At the same time, store the address when Master-MPU Error occurs in BUSnERRADD (n = 4, 5), store the RW information when Master-MPU Error occurs in BUSnERRRW (n = 4, 5), set 1 to MMERRSTAT bit of BUSnERRSTAT (n = 1 to 5, 10). Furthermore, NMI request or Reset request is generated according to the setting of the MMPUOAD.OAD. Since BUSnERRADD (n = 4, 5), BUSnERRRW (n = 4, 5), BUSnERRSTAT (n = 1 to 5, 10) are held until reset other than Bus error reset and Common memory error reset, or cleared by BUSnERRCLR (n = 1 to 5, 10), they can be confirmed in the NMI handler or after reset. NMI request is generated only at the first Master-MPU Error after reset or clearing BUSnERRSTAT (n = 1 to 5, 10). MMERRSTAT bit by BUSnERRCLR (n = 1 to 5, 10).

## (3) Illegal Access Error

According to [section 14.4.3. Conditions Leading to Illegal Address Access Errors](#), Illegal Access Error occurs. When Illegal Access Error is detected, an Error response is returned to the Master. When a master other than DAP caused Illegal address access error, store the Illegal Access Error occurrence address in BUSnERRADD (n = 4, 5), store the Illegal Access Error occurrence RW information in BUSnERRRW (n = 4, 5), set 1 to ILERRSTAT bit of BUSnERRSTAT (n = 1 to 5, 10). Furthermore, NMI request or Reset request is generated according to the setting of the BUSOAD.ILERRROAD. Since BUSnERRADD (n = 4, 5), BUSnERRRW (n = 4, 5), BUSnERRSTAT (n = 1 to 5, 10) are held until reset other than Bus error reset and Common memory error reset, or cleared by BUSnERRCLR (n = 1 to 5, 10), they can be confirmed in the BusFault Handler or the NMI handler. NMI request is generated only at the first Illegal Error after reset or clearing BUSnERRSTAT (n = 1 to 5, 10). ILERRSTAT bit by BUSnERRCLR (n = 1 to 5, 10).

## (4) Slave TrustZone Filter Error

When Slave TrustZone Filter Error is detected, an Error response is returned to the Master. When a master other than DAP caused Slave TrustZone Filter error, store the Slave TrustZone Filter error occurrence address in BMSAnERRADD (n = 4, 5), store the Slave TrustZone Filter error occurrence RW information in BMSAnERRRW (n = 4, 5), set 1 to SLERRSTAT bit of BUSnERRSTAT (n = 1 to 5, 10). Furthermore, NMI request or Reset request is generated according to the setting of the BUSOAD.SLERRROAD. Since BMSAnERRADD (n = 4, 5), BMSAnERRRW (n = 4, 5) and BUSnERRSTAT (n = 1 to 5, 10) are held until reset other than Bus error reset and Common memory error reset, or cleared by BUSnERRCLR (n = 1 to 5, 10), they can be confirmed in the NMI handler or after reset. NMI request is generated only at the first Slave TrustZone Filter Error after reset or clearing BUSnERRSTAT (n = 1 to 5, 10). SLERRSTAT bit by BUSnERRCLR (n = 1 to 5, 10).

## (5) Slave Bus Error

Slave Bus Error occurs in the slave. When Slave Bus Error is detected, an Error response is returned to the Master. When a master other than DAP caused Slave Bus error, store the Slave Bus error occurrence address in BUSnERRADD (n = 4, 5), store the Slave Bus error Occurrence RW information in BUSnERRRW (n = 4, 5), set 1 to SLERRSTAT bit of BUSnERRSTAT (n = 1 to 5, 10). Furthermore, NMI request or Reset request is generated according to the setting of the BUSOAD.SLERRROAD. Since BUSnERRADD (n = 4, 5), BUSnERRRW (n = 4, 5), BUSnERRSTAT (n = 1 to 5, 10) are held until reset other than Bus error reset and Common memory error reset, or cleared by BUSnERRCLR (n = 1 to 5, 10), they can be confirmed in the BusFault Handler or NMI handler. NMI request is generated only at the first Slave-Bus Error after reset or clearing BUSnERRSTAT (n = 1 to 5, 10). SLERRSTAT bit by BUSnERRCLR (n = 1 to 5, 10).

### 14.4.3 Conditions Leading to Illegal Address Access Errors

Illegal address access error by bus access occurs in the address space shown in [Table 14.15](#).

**Table 14.15 Conditions leading to illegal address access errors (1 of 3)**

Address		SLAVE bus	MASTER bus				
Start	End		CPU MAXI	CPU PAHB	DMAC/DTC	EDMAC	CEU
0x0000_0000	0x00FF_FFFF	CPU SAHBI	E	N/A	—	E	E
0x0100_0000	0x01FF_FFFF	Reserved	E	N/A	E	E	E
0x0200_0000	0x03FF_FFFF	FHBI	—	N/A	—	E	E

Table 14.15 Conditions leading to illegal address access errors (2 of 3)

Address		SLAVE bus	MASTER bus				
Start	End		CPU MAXI	CPU PAHB	DMAC/DTC	EDMAC	CEU
0x0400_0000	0x0FFF_FFFF	Reserved	E	N/A	E	E	E
0x1000_0000	0x10FF_FFFF	CPU SAHBI	E	N/A	—	E	E
0x1100_0000	0x11FF_FFFF	Reserved	E	N/A	E	E	E
0x1200_0000	0x13FF_FFFF	FHBI	—	N/A	—	E	E
0x1400_0000	0x1FFF_FFFF	Reserved	E	N/A	E	E	E
0x2000_0000	0x20FF_FFFF	CPU SAHBI	E	N/A	—	E	E
0x2100_0000	0x21FF_FFFF	Reserved	E	N/A	E	E	E
0x2206_0000	0x220D_FFFF	S1BI	—	N/A	—	—	—
0x220E_0000	0x25FF_FFFF	Reserved	E	N/A	E	E	E
0x2600_0000	0x2600_03FF	STBYSBI	—	N/A	—	—	E
0x2600_0400	0x26FF_FFFF	Reserved	E	N/A	E	E	E
0x2700_0000	0x2703_FFFF	FLBI	—	N/A	—	E	E
0x2704_0000	0x2FFF_FFFF	Reserved	E	N/A	E	E	E
0x3000_0000	0x30FF_FFFF	CPU SAHBI	E	N/A	—	E	E
0x3100_0000	0x31FF_FFFF	Reserved	E	N/A	E	E	E
0x3206_0000	0x320D_FFFF	S1BI	—	N/A	—	—	—
0x320E_0000	0x35FF_FFFF	Reserved	E	N/A	E	E	E
0x3600_0000	0x3600_03FF	STBYSBI	—	N/A	—	—	E
0x3600_0400	0x36FF_FFFF	Reserved	E	N/A	E	E	E
0x3700_0000	0x3703_FFFF	FLBI	—	N/A	—	E	E
0x3704_0000	0x3FFF_FFFF	Reserved	E	N/A	E	E	E
0x4000_0000	0x4000_0FFF	PSBI	N/A	—	—	E	E
0x4000_1000	0x4000_1FFF	Reserved	N/A	E*1	E	E	E
0x4000_2000	0x4000_4FFF	PSBI	N/A	—	—	E	E
0x4000_5000	0x4000_5FFF	Reserved	N/A	E*1	E	E	E
0x4000_6000	0x4000_6FFF	PSBI	N/A	—	—	E	E
0x4000_7000	0x4000_7FFF	Reserved	N/A	E*1	E	E	E
0x4000_8000	0x4000_8FFF	PSBI	N/A	—	—	E	E
0x4000_9000	0x4000_9FFF	Reserved	N/A	E*1	E	E	E
0x4000_A000	0x4000_AFFF	PSBI	N/A	—	—	E	E
0x4000_B000	0x4000_BFFF	Reserved	N/A	E*1	E	E	E
0x4000_C000	0x4000_CFFF	PSBI	N/A	—	E	E	E
0x4000_D000	0x4000_EFFF	Reserved	N/A	E*1	E	E	E
0x4000_F000	0x4001_CFFF	PSBI	N/A	—	—	E	E
0x4001_D000	0x4001_DFFF	Reserved	N/A	E*1	E	E	E
0x4001_E000	0x4001_EFFF	PSBI	N/A	—	—	E	E
0x4001_F000	0x400F_FFFF	Reserved	N/A	E*1	E	E	E
0x4010_0000	0x4011_FFFF	FLBI	N/A	—	—	E	E
0x4012_0000	0x401F_FFFF	Reserved	N/A	E*1	E	E	E
0x4020_0000	0x402F_FFFF	PBBI	N/A	—	—	E	E

**Table 14.15 Conditions leading to illegal address access errors (3 of 3)**

Address		SLAVE bus	MASTER bus				
Start	End		CPU MAXI	CPU PAHB	DMAC/DTC	EDMAC	CEU
0x4030_0000	0x403F_FFFF	PABI	N/A	—	—	E	E
0x4040_0000	0x404F_FFFF	PIBI	N/A	—	—	E	E
0x4050_0000	0x4FFF_FFFF	Reserved	N/A	E*1	E	E	E
0x5000_0000	0x5000_0FFF	PSBI	N/A	—	—	E	E
0x5000_1000	0x5000_1FFF	Reserved	N/A	E*1	E	E	E
0x5000_2000	0x5000_4FFF	PSBI	N/A	—	—	E	E
0x5000_5000	0x5000_5FFF	Reserved	N/A	E*1	E	E	E
0x5000_6000	0x5000_6FFF	PSBI	N/A	—	—	E	E
0x5000_7000	0x5000_7FFF	Reserved	N/A	E*1	E	E	E
0x5000_8000	0x5000_8FFF	PSBI	N/A	—	—	E	E
0x5000_9000	0x5000_9FFF	Reserved	N/A	E*1	E	E	E
0x5000_A000	0x5000_AFFF	PSBI	N/A	—	—	E	E
0x5000_B000	0x5000_BFFF	Reserved	N/A	E*1	E	E	E
0x5000_C000	0x5000_CFFF	PSBI	N/A	—	E	E	E
0x5000_D000	0x5000_EFFF	Reserved	N/A	E*1	E	E	E
0x5000_F000	0x5000_FFFF	PSBI	N/A	—	—	E	E
0x5001_0000	0x5001_CFFF	PSBI	N/A	—	—	E	E
0x5001_D000	0x5001_DFFF	Reserved	N/A	E*1	E	E	E
0x5001_E000	0x5001_EFFF	PSBI	N/A	—	—	E	E
0x5001_F000	0x500F_FFFF	Reserved	N/A	E*1	E	E	E
0x5010_0000	0x5011_FFFF	FLBI	N/A	—	—	E	E
0x5012_0000	0x501F_FFFF	Reserved	N/A	E*1	E	E	E
0x5020_0000	0x502F_FFFF	PBBI	N/A	—	—	E	E
0x5030_0000	0x503F_FFFF	PABI	N/A	—	—	E	E
0x5040_0000	0x504F_FFFF	PIBI	N/A	—	—	E	E
0x5050_0000	0x5FFF_FFFF	Reserved	N/A	E*1	E	E	E
0x7000_0000	0x7FFF_FFFF	Reserved	E	N/A	E	E	E
0x8000_0000	0x9FFF_FFFF	EOBI	—	E	—	—	—
0xA000_0000	0xFFFF_FFFF	Reserved	E	N/A	E	E	E

Note: "E" : A bus error is occurred.

"N/A" : Transfer does not occur.

"—" : A bus error has not occurred. Even if there is reserved area, a bus error has not occurred.

Do not access the reserved area in FLB and S1B. If accessed, a slave TZF error may occur.

Note 1. If CPU issues bufferable write accesses, a bufferable write access error occur instead of an illegal address access error.

#### 14.4.4 Timeout

As for some peripheral module, a timeout error can occur in the module-stop function.

When there is no response from the slave for a certain time period, a timeout error is detected.

Timeout error is returned to the request master by Error response of the AHB-Lite protocol.

### 14.5 References

1. ARM Limited, *Arm v8-M Architecture Reference Manual*(ARM DDI0553B.g)



2. ARM Limited, *Arm<sup>®</sup> Cortex<sup>®</sup>-M85 Processor Integration and Implementation Manual Revision:r0p2* (ARM 101925\_0002\_05\_en)
3. ARM Limited, *Arm<sup>®</sup> Cortex<sup>®</sup>-M85 Processor Technical Reference Manual Revision:r0p2* (ARM 101924\_0002\_05\_en)
4. ARM Limited, *Arm<sup>®</sup> Cortex<sup>®</sup>-M85 Processor User Guide Reference Material Revision:r0p2* (ARM 101927\_0002\_05\_en)
5. ARM Limited, *ARM AMBA 5 AHB Protocol Specification AHB5, AHB-Lite* (ARM IHI 0033B.b)
6. ARM Limited, *ARM AMBA AXI and ACE Protocol Specification AXI3, AXI4, and AXI4-Lite, ACE and ACE-Lite* (ARM IHI 0022H.c)
7. ARM Limited, *ARM AMBA APB Protocol Specification Version: 2.0* (ARM IHI 0024D)

## 15. Memory Protection Unit (MPU)

### 15.1 Overview

All bus masters have Memory Protection Units (MPUs).

[Table 15.1](#) lists the MPU specifications, and [Table 15.2](#) shows the behavior on detection of each MPU error.

**Table 15.1 MPU specifications**

Classification	Module/Function	Specifications
Memory protection	Arm MPU	Memory protection function for the CPU: <ul style="list-style-type: none"> <li>• CPU: Secure MPU 8 regions and Non-secure MPU 8 regions</li> </ul>
	Bus master MPU	Memory protection function for each bus master except for the CPU: <ul style="list-style-type: none"> <li>• DMAC (DMAC/DTC): 8 regions</li> <li>• EDMAC (Ether-DMAC): 4 regions</li> <li>• CEU: 2 regions</li> </ul>

**Table 15.2 Behavior on MPU Error detection**

MPU type	Access permissions setting	Boundary address setting minimum unit	Error response for the MPU error notification	Bus access at error detection	Hold the information of error access
Arm MPU	Read access Write access Execution	32 bytes	Supported*1	<ul style="list-style-type: none"> <li>• Incorrectly write access</li> <li>• Incorrectly read access</li> </ul>	Hold in CPU
Bus master MPU	Read access Write access Privileged access (DMAC/DTC only)	DMAC: 32 bytes EDMAC: 32 bytes CEU: 4 KB	Supported	<ul style="list-style-type: none"> <li>• Write access ignored</li> <li>• Read access is read as 0</li> </ul>	Hold

Note 1. A privileged DAP request through the unprivileged debug extension mechanism is demoted to an unprivileged access and is subject to MPU checks. Both privileged and unprivileged requests are subject to MPU checks.

For information on error access for the Arm MPU, see [section 15.4. References](#). For information on error access for other MPUs, see [section 15.3.1. Register Descriptions](#) and [section 14.4. Bus Error Monitoring Section](#).

### 15.2 Arm MPU

The Arm MPU provides full support for:

- CPU: 8 secure regions and 8 non-secure regions
- Access permissions
- Export of memory attributes to the system

Arm MPU mismatches and permission violations invoke the programmable-priority MemManage fault (Hard Fault) handler. For details, see [section 15.4. References](#).

### 15.3 Bus master MPU

The bus master MPU monitors the addresses accessed by the bus masters in the entire address space (0x0000\_0000 to 0xFFFF\_FFFF). The access control information can be set up to:

- 8 regions in DMAC/DTC
- 4 regions in EDMAC
- 2 regions in CEU

The bus master MPU monitors access to each region based on these settings.

If accesses violate the access permissions that are configured in bus master MPU, the bus master MPU returns an error response. For information on error access, see [section 15.3.1. Register Descriptions](#) and [section 14.4.2. Operations When a Bus Error Occurs](#).

[Table 15.3](#) lists the specifications of the bus master MPU.

**Table 15.3 Bus master MPU specifications**

Parameter	Description
Master groups	(Group name: Corresponding bus master) <ul style="list-style-type: none"> <li>• DMAC: DMAC/DTC</li> <li>• EDMAC: Ether-DMAC</li> <li>• CEU: CEU</li> </ul>
Protected regions	0x0000_0000 to 0xFFFF_FFFF
Number of regions	<ul style="list-style-type: none"> <li>• DMAC: 8 regions</li> <li>• EDMAC: 4 regions</li> <li>• CEU: 2 regions</li> </ul>
Address specification for individual regions	<ul style="list-style-type: none"> <li>• Region start and end addresses configurable</li> </ul>
Enable or disable setting for memory protection in individual regions	<ul style="list-style-type: none"> <li>• Settings enabled or disabled for the associated region</li> </ul>
Access-control settings for individual regions	<ul style="list-style-type: none"> <li>• Permission to read and write</li> <li>• Permission to unprivileged access</li> </ul>
Operation on error detection	<ul style="list-style-type: none"> <li>• Reset or error response</li> </ul>
Register protection	<ul style="list-style-type: none"> <li>• Register can be protected from illegal writes</li> </ul>

### 15.3.1 Register Descriptions

Register writes processing should be performed after stopping the corresponding master group bus access.

SA (Secure Attribution) is set in each register and the SA is used to determine whether the target register is a secure register or a non-secure register.

#### 15.3.1.1 MMPUSARA : Master Memory Protection Unit Security Attribution Register A

Base address: CPSCU = 0x4000\_8000  
 CPSCU\_NS = 0x5000\_8000

Offset address: 0x130

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	MMPUASAn[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	MMPUASAn[7:0]	MMPUA Security Attribution n (n = 0 to 7) 0: Secure 1: Non-secure	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1  
 P-TYPE-1

#### MMPUASAn[7:0] bits (MMPUA Security Attribution n (n = 0 to 7))

Security attributes of bus master MPU Region Setting register. The target registers are as follows:

- MMPUSDMAcN (n = 0 to 7)
- MMPUEDMAcN (n = 0 to 7)
- MMPUACDMAcN (n = 0 to 7)

### 15.3.1.2 MMPUSARB : Master Memory Protection Unit Security Attribution Register B

Base address: CPSCU = 0x4000\_8000  
 CPSCU\_NS = 0x5000\_8000

Offset address: 0x134

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MMPU BSA8	—	—	—	—	—	—	—	MMPU BSA0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MMPUBSA0	MMPUB Security Attribution 0 0: Secure 1: Non-secure	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	MMPUBSA8	MMPUB Security Attribution 8 0: Secure 1: Non-secure	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1  
 P-TYPE-1

#### MMPUBSA0 bit (MMPUB Security Attribution 0)

Security attributes of DMAC MPU Enable Setting register. The target registers are as follows:

- MMPUENDMAC
- MMPUENPTDMAC

#### MMPUBSA8 bit (MMPUB Security Attribution 8)

Security attributes of MPU Operation After Detection Setting register. The target registers are as follows:

- MMPUOAD
- MMPUOADPT

### 15.3.1.3 MMPUSDMACn : MPU Start Address Register for DMAC (n = 0 to 7)

Base address: RMPU = 0x4000\_0000  
 RMPU\_NS = 0x5000\_0000

Offset address: 0x0204 + 0x010 × n

Bit position: 31 0

Bit field:	[Empty box for bit field]															
------------	---------------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Value after reset: x 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Region start address register Address where the region starts, for use in region determination. The starting address of the MPU area must be set in the range of 0x0000_0000 to 0xFFFF_FFE0. Writing is ignored for bit 0 to bit 4. These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3  
P-TYPE-2

The MMPUSDMACn (n = 0 to 7) register specifies the start address where the region starts.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

Regions set by MMPUSDMACn (n = 0 to 7), MMPUEDMACn (n = 0 to 7) and MMPUACDMACn (n = 0 to 7) registers, can be set for a secure access or a non-secure access with the MMPUSARA register. If the corresponding MMPUSARA.MMPUASAn (n = 0 to 7) bit is set to 1, it is only possible to use that region with a non-secure access. On the other hand, if the corresponding MMPUSARA.MMPUASAn (n = 0 to 7) bit is set to 0, it is only possible to use that region with a secure access.

15.3.1.4 MMPUSEDMACn : MPU Start Address Register for EDMAC (n = 0 to 3)

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0604+ 0x010 × n

Bit position: 31 0



Value after reset: x 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Region start address register for EDMAC Address where the region starts, for use in region determination. The starting address of the MPU area must be set in the range of 0x0000_0000 to 0xFFFF_FFE0. Writing is ignored for bit 0 to bit 4. These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3  
P-TYPE-2

The MMPUSEDMACn (n = 0 to 3) register specifies the start address where the region starts.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.5 MMPUSCEUn : MPU Start Address Register for CEU (n = 0 to 1)

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0E04+ 0x010 × n

Bit position: 31 0



Value after reset: x 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Region start address register for CEU Address where the region starts, for use in region determination. The starting address of the MPU area must be set in the range of 0x0000_0000 to 0xFFFF_F000. Writing is ignored for bit 0 to bit 11. These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3  
P-TYPE-2

The MMPUSCEUn (n = 0 to 1) register specifies the start address where the region starts.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.6 MMPUEDMACn : MPU End Address Register for DMAC (n = 0 to 7)

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0208+ 0x010 × n

Bit position: 31 0



Value after reset: x 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	n/a	Region end address register Address where the region end, for use in region determination. The ending address of the MPU area must be set in the range of 0x0000_001F to 0xFFFF_FFFF. Writing is ignored for bit 0 to bit 4. These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3  
P-TYPE-2

The MMPUEDMACn (n = 0 to 7) register specifies the end address where the region ends.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.7 MMPUEEDMACn : MPU End Address Register for EDMAC (n = 0 to 3)

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0608+ 0x010 × n

Bit position: 31 0



Value after reset: x 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	n/a	Region end address register for EDMAC Address where the region ends, for use in region determination. The ending address of the MPU area must be set in the range of 0x0000_001F to 0xFFFF_FFFF. Writing is ignored for bit 0 to bit 4. These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3  
P-TYPE-2

The MMPUEEDMACn (n = 0 to 3) register specifies the end address where the region ends.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

15.3.1.8 MMPUECEUn : MPU End Address Register for CEU (n = 0 to 1)

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0E08+ 0x010 × n

Bit position: 31 0



Value after reset: x 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
31:0	n/a	Region end address register for CEU Address where the region ends, for use in region determination. The ending address of the MPU area must be set in the range of 0x0000_0FFF to 0xFFFF_FFFF. Writing is ignored for bit 0 to bit 11. These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3  
P-TYPE-2

The MMPUECEUn (n = 0 to 1) register specifies the end address where the region ends.

This register requires word access. Byte access and halfword access is prohibited. When byte access and halfword access is executed, operation is not guaranteed.

### 15.3.1.9 MMPUACDMACn : MPU Access Control Register for DMAC (n = 0 to 7)

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0200+ 0x010 × n

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PP	WP	RP	ENAB LE
------------	---	---	---	---	---	---	---	---	---	---	---	---	----	----	----	------------

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: DMAC region n unit is disabled. 1: DMAC region n unit is enabled.	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
3	PP	Privilege protection 0: Unprivileged access permission 1: Unprivileged access protection	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3  
P-TYPE-2

#### ENABLE bit (Region Enable)

The ENABLE bit enables or disables the DMAC region n (n = 0 to 7) unit.

When the ENABLE bit is set to 1, the RP bit, the WP bit and the PP bit are enabled for permit or protect access to the region that is set in MMPUSDMACn (n = 0 to 7) and MMPUEDMACn (n = 0 to 7).

When the ENABLE bit is set to 0, the access to DMAC region n (n = 0 to 7) is not specified.

#### RP bit (Read protection)

The RP bit enables or disables read protection for DMAC region n (n = 0 to 7).

When the ENABLE bit is set to 1, the RP bit is available.

#### WP bit (Write protection)

The WP bit enables or disables write protection for DMAC region n (n = 0 to 7).

When the ENABLE bit is set to 1, the WP bit is available.

**PP bit (Privilege protection)**

The PP bit enables or disables unprivileged access protection for DMAC region n (n = 0 to 7).

When the ENABLE bit is set to 1, the PP bit is available.

**15.3.1.10 MMPUACEDMACn : MPU Access Control Register for EDMAC (n = 0 to 3)**

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0600+ 0x010 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	WP	RP	ENAB LE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: EDMAC region n unit is disabled. 1: EDMAC region n unit is enabled.	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3  
P-TYPE-2

**ENABLE bit (Region Enable)**

The ENABLE bit enables or disables the EDMAC region n (n = 0 to 3) unit.

When the ENABLE bit is set to 1, the RP bit and the WP bit are enabled for permit or protect access to the region that is set in MMPUSEDMACn (n = 0 to 3) and MMPUEEDMACn (n = 0 to 3).

When the ENABLE bit is set to 0, the access to EDMAC region n (n = 0 to 3) is not specified.

**RP bit (Read protection)**

The RP bit enables or disables read protection for EDMAC region n (n = 0 to 3).

When the ENABLE bit is set to 1, the RP bit is available.

**WP bit (Write protection)**

The WP bit enables or disables write protection for EDMAC region n (n = 0 to 3).

When the ENABLE bit is set to 1, the WP bit is available.

**15.3.1.11 MMPUACCEUn : MPU Access Control Register for CEU (n = 0, 1)**

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0E00+ 0x010 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	WP	RP	ENAB LE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Bit	Symbol	Function	R/W
0	ENABLE	Region Enable 0: CEU region n unit is disabled. 1: CEU region n unit is enabled.	R/W
1	RP	Read protection 0: Read permission 1: Read protection	R/W
2	WP	Write protection 0: Write permission 1: Write protection	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3  
P-TYPE-2

**ENABLE bit (Region Enable)**

The ENABLE bit enables or disables the CEU region n (n = 0, 1) unit.

When the ENABLE bit is set to 1, the RP bit and the WP bit are enabled for permit or protect access to the region that is set in MMPUSCEUn (n = 0, 1) and MMPUECEUn (n = 0, 1).

When the ENABLE bit is set to 0, the access to CEU region n (n = 0, 1) is not specified.

**RP bit (Read protection)**

The RP bit enables or disables read protection for CEU region n (n = 0, 1).

When the ENABLE bit is set to 1, the RP bit is available.

**WP bit (Write protection)**

The WP bit enables or disables write protection for CEU region n (n = 0, 1).

When the ENABLE bit is set to 1, the WP bit is available.

**15.3.1.12 MMPUENDMAC : MPU Enable Register for DMAC**

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0100

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	KEY[7:0]														—	—	—	—	—	—	—	—	ENAB LE
------------	----------	--	--	--	--	--	--	--	--	--	--	--	--	--	---	---	---	---	---	---	---	---	------------

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	ENABLE	Bus master MPU of DMAC Enable 0: Bus master MPU of DMAC is disabled. 1: Bus master MPU of DMAC is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the ENABLE bit.	W

Note: S-TYPE-3  
P-TYPE-2

Note: It is necessary to write by halfword access. Byte-write access is prohibited.

When byte-write access is executed, operation is not guaranteed.

**ENABLE bit (Bus master MPU of DMAC Enable)**

The ENABLE bit enables or disables the bus master MPU function for DMAC.

When the ENABLE bits is set to 1, MMPUACDMACn (n = 0 to 7) is available.

When the ENABLE bits is set to 0, MMPUACDMACn (n = 0 to 7) is unavailable, including permission for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When the ENABLE bit is set, write 0xA5 in KEY[7:0] at the same time.

### KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing of the ENABLE.

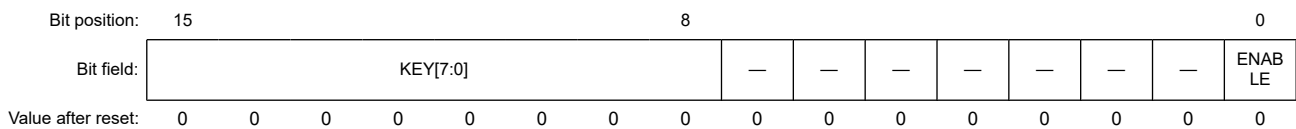
When writing the ENABLE bit, write 0xA5 in KEY[7:0] at the same time.

When a value other than 0xA5 is written in KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

### 15.3.1.13 MMPUENEDMAC : MPU Enable Register for EDMAC

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0500



Bit	Symbol	Function	R/W
0	ENABLE	Bus master MPU of EDMAC Enable 0: Bus master MPU of EDMAC is disabled. 1: Bus master MPU of EDMAC is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the ENABLE bit.	W

Note: S-TYPE-3  
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

### ENABLE bit (Bus master MPU of EDMAC Enable)

The ENABLE bit enables or disables the bus master MPU function of each master group.

When the ENABLE bit is set to 1, MMPUACEDMACn (n = 0 to 3) is available.

When the ENABLE bit is set to 0, MMPUACEDMACn (n = 0 to 3) is unavailable, and permission for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When the ENABLE bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

### KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the ENABLE bit.

When writing the ENABLE bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

### 15.3.1.14 MMPUENCEU : MPU Enable Register for CEU

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0D00



Bit	Symbol	Function	R/W
0	ENABLE	Bus master MPU of CEU Enable 0: Bus master MPU of CEU is disabled. 1: Bus master MPU of CEU is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the ENABLE bit.	W

Note: S-TYPE-3  
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

#### ENABLE bit (Bus master MPU of CEU Enable)

The ENABLE bit enables or disables the bus master MPU function of each master group.

When the ENABLE bit is set to 1, MMPUACCEUn (n = 0, 1) is available.

When the ENABLE bit is set to 0, MMPUACCEUn (n = 0, 1) is unavailable, and permission for all regions.

The bus master MPU function of each master group sets the ENABLE bit.

When the ENABLE bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

#### KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the ENABLE bit.

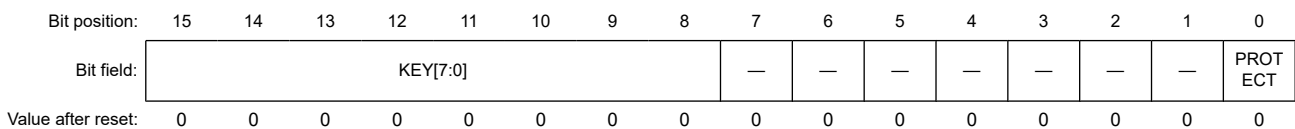
When writing the ENABLE bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

### 15.3.1.15 MMPUENPTDMAC : MPU Enable Protect Register for DMAC

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0104



Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENDMAC register write is possible. 1: MMPUENDMAC register write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the PROTECT bit.	W

Note: S-TYPE-3  
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

### PROTECT bit (Protection of register)

The PROTECT bit controls protection of MMPUENDMAC register.

When the PROTECT bit is set, write 0xA5 in KEY[7:0] at the same time.

### KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing of the PROTECT.

When writing the PROTECT bit, write 0xA5 in KEY[7:0] at the same time.

When a value other than 0xA5 is written in KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

#### 15.3.1.16 MMPUENPTEDMAC : MPU Enable Protect Register for EDMAC

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0504

Bit position: 15 8 0

Bit field:	KEY[7:0]	—	—	—	—	—	—	—	PROTECT
Value after reset:	0 0 0 0 0 0 0 0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENEDMAC register write is possible. 1: MMPUENEDMAC register write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3  
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

### PROTECT bit (Protection of register)

The PROTECT bit controls protection of MMPUENEDMAC register.

When the PROTECT bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

### KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the PROTECT bit.

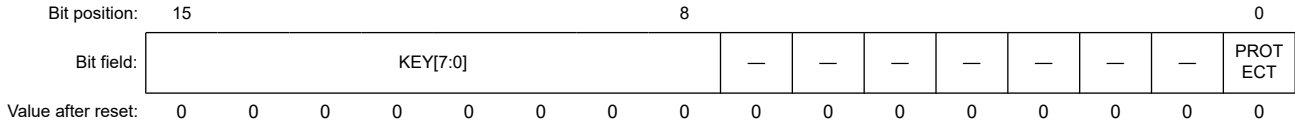
When writing the PROTECT bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

### 15.3.1.17 MMPUENPTCEU : MMPU Enable Protect Register for CEU

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0D04



Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUENPTCEU register write is possible. 1: MMPUENPTCEU register write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3  
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

#### PROTECT bit (Protection of register)

The PROTECT bit controls protection of MMPUENPTCEU register.

When the PROTECT bit is set, write 0xA5 to the KEY[7:0] bits at the same time.

#### KEY[7:0] bits (Key Code)

The KEY[7:0] bits are used to enable or disable writing to the PROTECT bit.

When writing the PROTECT bit, write 0xA5 to the KEY[7:0] bits at the same time.

When a value other than 0xA5 is written to the KEY[7:0] bits, the ENABLE bit is not updated. The KEY[7:0] bits are always read as 0x00.

### 15.3.1.18 MMPURPTDMAC : MMPU Regions Protect Register for DMAC Non-secure

Base address: RMPU\_NS = 0x5000\_0000

Offset address: 0x0108



Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for DMAC write is possible. 1: Bus master MPU register for DMAC write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the PROTECT bit.	W

Note: S-TYPE-7  
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

### PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MMPURPTDMAC.PROTECT controls the following registers:

- MMPUSDMAC<sub>n</sub> (n = 0 to 7) of non-secure
- MMPUEDMAC<sub>n</sub> (n = 0 to 7) of non-secure
- MMPUACDMAC<sub>n</sub> (n = 0 to 7) of non-secure

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits, using halfword access.

### KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written in KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

#### 15.3.1.19 MMPURPTDMAC\_SEC : MPU Regions Protect register for DMAC Secure

Base address: RMPU = 0x4000\_0000

Offset address: 0x010C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for DMAC Secure write is possible. 1: Bus master MPU register for DMAC Secure write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the PROTECT bit.	W

Note: S-TYPE-6  
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

### PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MMPURPTDMAC\_SEC.PROTECT controls the following registers:

- MMPUSDMAC<sub>n</sub> (n = 0 to 7) of Secure
- MMPUEDMAC<sub>n</sub> (n = 0 to 7) of Secure
- MMPUACDMAC<sub>n</sub> (n = 0 to 7) of Secure

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits, using halfword access.

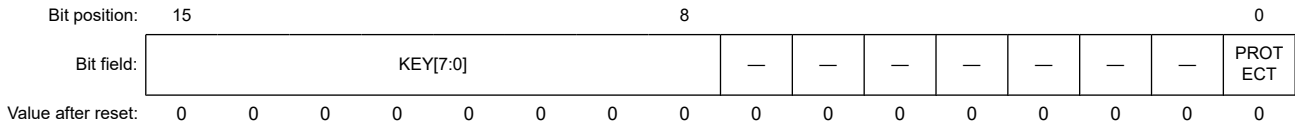
### KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written in KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

### 15.3.1.20 MMPURPTEDMAC : MPU Regions Protect Register for EDMAC

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0508



Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for EDMAC write is possible. 1: Bus master MPU register for EDMAC write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3  
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

#### PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPURPTEDMAC.PROTECT controls the following registers:

- MMPUSEDMAC<sub>n</sub> (n = 0 to 3)
- MMPUEEDMAC<sub>n</sub> (n = 0 to 3)
- MMPUACEDMAC<sub>n</sub> (n = 0 to 3)

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

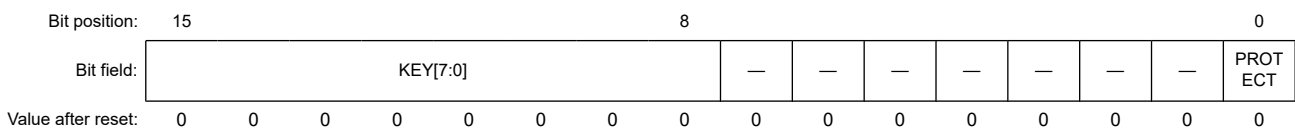
#### KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

### 15.3.1.21 MMPURPTCEU : MPU Regions Protect Register for CEU

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0D08



Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: Bus master MPU register for CEU write is possible. 1: Bus master MPU register for CEU write is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3  
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

### PROTECT bit (Protection of register)

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPURPTCEU.PROTECT controls the following registers:

- MMPUSCEUn (n = 0, 1)
- MMPUECEUn (n = 0, 1)
- MMPUACCEUn (n = 0, 1)

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

### KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When a value other than 0xA5 is written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

#### 15.3.1.22 MMPUOAD : MPU Operation After Detection Register

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Bit field:	KEY[7:0]														—	—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						

Bit	Symbol	Function	R/W
0	OAD	Operation after detection 0: NMI 1: Reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code This bit is used to enable or disable writing of the OAD bit.	R/W

Note: S-TYPE-3  
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

### OAD bit (Operation after detection)

The OAD bit specify operation when the access violation is detected.

When OAD bit is 0, error response is returned and NMI is generated.

When OAD bit is 1, reset request is generated.

When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.



**KEY[7:0] bits (Key Code)**

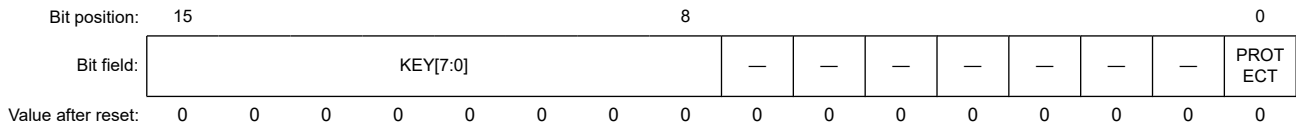
The KEY[7:0] bits enable or disable writing to the OAD bit. When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits. When other values are written, the OAD bit is not updated.

The KEY[7:0] bits always read as 0x00.

**15.3.1.23 MMPUOADPT : MPU Operation After Detection Protect Register**

Base address: RMPU = 0x4000\_0000  
RMPU\_NS = 0x5000\_0000

Offset address: 0x0004



Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MMPUOAD register writing is possible. 1: MMPUOAD register writing is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writing to the PROTECT bit.	W

Note: S-TYPE-3  
P-TYPE-2

Note: It is necessary to write by halfword access.

Byte-write access is prohibited. When byte-write access is executed, operation is not guaranteed.

**PROTECT bit (Protection of register)**

The PROTECT bit enables or disables writing to the associated registers to be protected.

MMPUOAD.PROTECT controls the following registers:

- MMPUOAD

When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

**KEY[7:0] bits (Key Code)**

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. When other values are written to the KEY[7:0] bits, the PROTECT bit is not updated. The KEY[7:0] bits are always read as 0x00.

**15.3.2 Operation****15.3.2.1 Memory protection**

The bus master MPU monitors memory access using control settings made individually for the access control regions. If accesses violate the access permissions that are configured in bus master MPU, the bus master MPU generates a memory protection error.

Bus master MPU has master groups of DMAC, EDMAC and CEU. The memory protection function checks the address of the bus for a unified master group, and blocks illegal access of the master group to the protected region by bus master MPU.

The region setting registers of the bus master MPU for DMAC can be set for secure master and non-secure master using the MMPUSARA register. The MPU region setting with MMPUSARA.MMPUASAn = 0 applies only to access from the secure master, and the MPU region setting with MMPUSARA.MMPUASAn = 1 applies only to access from the non-secure master. [Figure 15.1](#) shows an example of access authorization for the secure master and non-secure master when both secure and non-secure MPU settings are configured.

	Memory attributes for Secure DMAC		Memory attributes for Non-secure DMAC	Secure DMAC access authorization	Non-secure DMAC access authorization
Non-secure alias region	Protect region		Protect region	All access blocked	All access blocked
	Region 1 (MMPUASA1 = Secure) Read/Write permit Unprivileged permit		Region 2 (MMPUASA2 = Non-secure) Read only (Write protect) Unprivileged protect	All access permitted (No effect from Region 2)	Unprivileged Read access blocked Unprivileged Write access blocked Privileged Read access permitted Privileged Write access blocked (No effect from Region 1)
				All access blocked	All access blocked
Secure alias region	Protect region		Protect region	All access blocked	All access protected (Security violation)
	Region0 (MMPUASA0 = Secure) Read/Write permit Unprivileged protect			Unprivileged R/W access blocked Privileged R/W access permitted	All access blocked (Security violation) (No effect from Region 0)

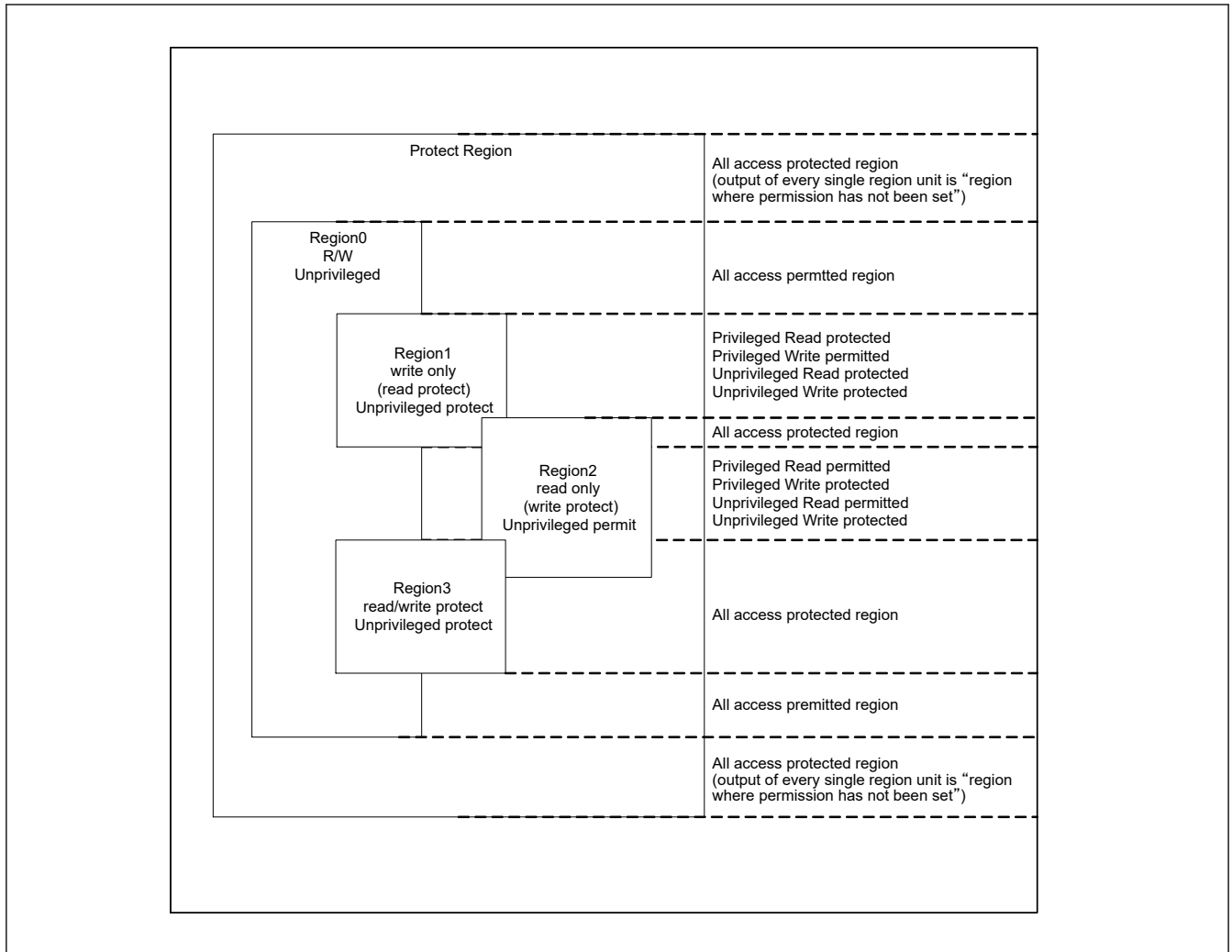
**Figure 15.1 Example of access authority when both secure and non-secure region of MPUs are set**

Bus master MPU permit of all regions after reset. If MMPUENXXXX.ENABLE is 1 and there is no MPU region setting for the corresponding bus master, all regions are protected. (XXXX = Master Group name)

Figure 15.2 shows the access permission or protection by the overlapping bus master MPU regions.

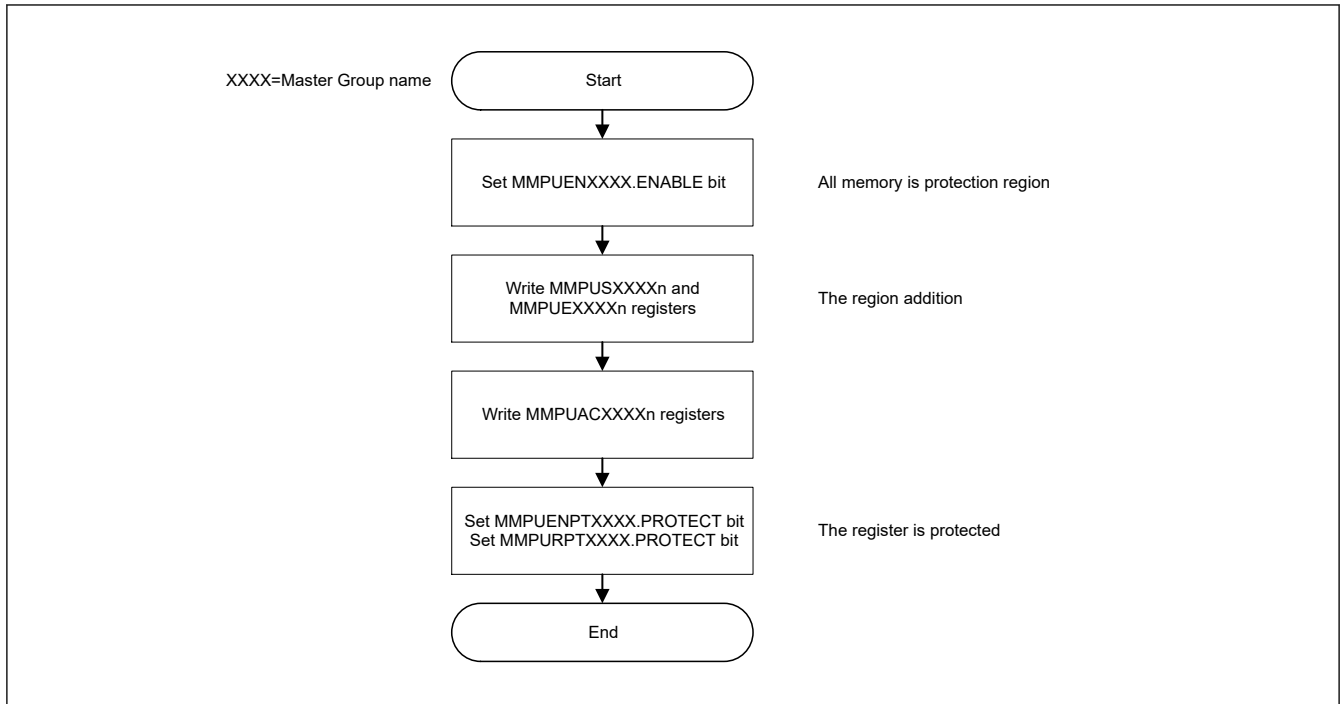
Access control for the overlapping regions is as follows:

- The region is handled as a protected region when output of one or more region units is a protected region
- The region is handled as a protected region when output of all region units is outside of the regions
- Other cases are handled as permitted regions.



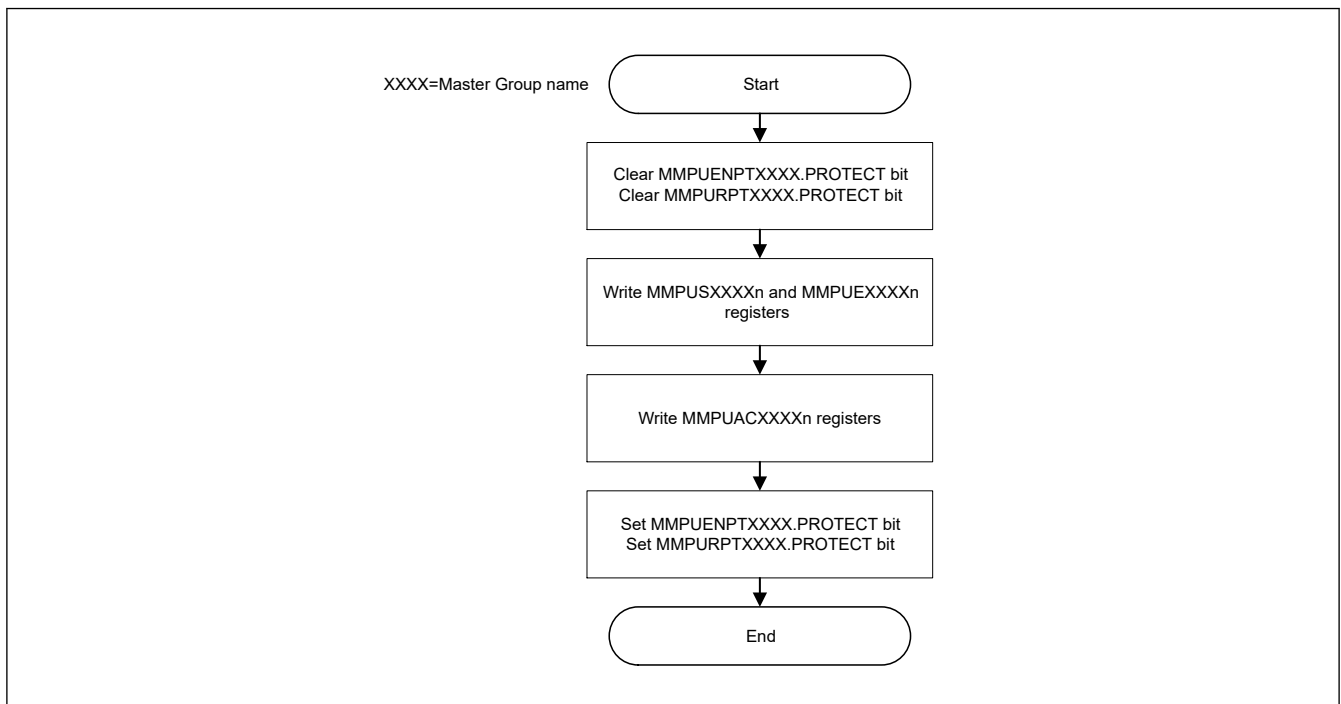
**Figure 15.2 Access permission or protection by overlap of the bus master MPU regions**

Figure 15.3 shows the register setting flow after reset. During this register setting, stop the bus master except the CPU.



**Figure 15.3 Register setting flow after reset**

Figure 15.4 shows the register setting flow for adding regions. During this register setting, stop the master except the CPU.



**Figure 15.4 Register setting flow for region addition**

### 15.3.2.2 Protecting the registers

Registers related to the bus master MPU can be protected with the PROTECT bit as shown in Table 15.4.

**Table 15.4 PROTECT bit and protect target registers (1 of 2)**

PROTECT bit	Protect target registers
MMPUENPTDMAC.PROTECT	MMPUENDMAC

**Table 15.4 PROTECT bit and protect target registers (2 of 2)**

PROTECT bit	Protect target registers
MMPUENPTEDMAC.PROTECT	MMPUENEDMAC
MMPUENPTCEU.PROTECT	MMPUENCEU
MMPURPTDMAC.PROTECT	The following registers set to Non-secure by MMPUSARA.MMPUASAn (n = 0 to 7). MMPUSDMACn (n = 0 to 7) MMPUEDMACn (n = 0 to 7) MMPUACDMACn (n = 0 to 7)
MMPURPTDMAC_SEC.PROTECT	The following registers set to Secure by MMPUSARA.MMPUASAn (n = 0 to 7). MMPUSDMACn (n = 0 to 7) MMPUEDMACn (n = 0 to 7) MMPUACDMACn (n = 0 to 7)
MMPURPTEDMAC.PROTECT	MMPUSEDMACn (n = 0 to 3) MMPUEEDMACn (n = 0 to 3) MMPUACEDMACn (n = 0 to 3)
MMPURPTCEU.PROTECT	MMPUSCEUn (n = 0 to 1) MMPUECEUn (n = 0 to 1) MMPUACCEUn (n = 0 to 1)
MMPUOADPT.PROTECT	MMPUOAD.OAD

### 15.3.2.3 Memory Protection error

If access to a protected region is detected, the bus master MPU generates an error. Set the MMPUOAD bit to select whether the error is reported as a non-maskable interrupt or a reset.

See [section 14, Buses](#) for details of bus master MPU error.

The non-maskable interrupt status is indicated in ICU.NMISR.BUSST. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#). The reset status is indicated in SYSC.RSTSR1.BUSRF. For details, see [section 5, Resets](#).

## 15.4 References

1. *ARM® v8-M Architecture Reference Manual* (ARM DDI0553B.g)
2. *ARM® Cortex®-M85 Processor Technical Reference Manual*, *ARM® Cortex®-M33 Processor Technical Reference Manual* (ARM 101924\_0002\_05\_en, ARM 100230\_0004\_00\_en)
3. *ARM® Cortex®-M85 Processor User Guide Reference Material*, *ARM® Cortex®-M33 Processor User Guide Reference Material* (ARM 101927\_0002\_05\_en, ARM 100234\_0002\_00\_en)
4. *ARM® Cortex®-M85 Processor Integration and Implementation Manual*, *ARM® Cortex®-M33 Processor Integration and Implementation Manual* (ARM 101925\_0002\_05\_en, ARM 100323\_0002\_00\_en)

## 16. DMA Controller (DMAC)

### 16.1 Overview

The 8-channel direct memory access controller (DMAC) that can transfer data without intervention from the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

Table 16.1 lists the DMAC specifications, and Figure 16.1 shows a block diagram of the DMAC.

**Table 16.1 DMAC specifications (1 of 2)**

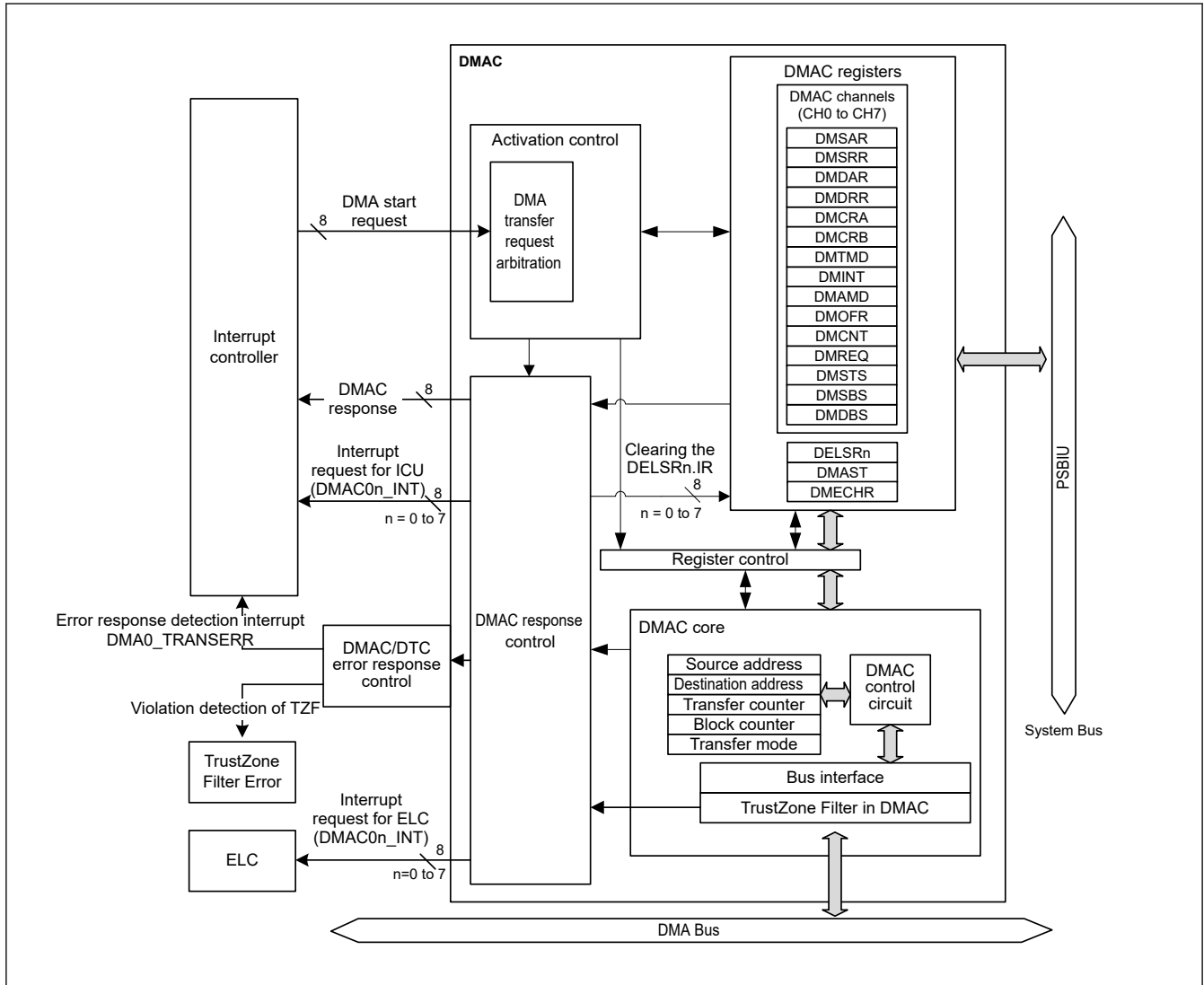
Item		Description
Number of channels		8 channels (DMACn (n = 0 to 7))
Transfer space		4 GB (0x00000000 to 0xFFFFFFFF excluding reserved areas)
Maximum transfer volume		64 M data (Maximum number of transfers in block transfer mode: 1,024 data × 65,536 blocks)
DMAC activation source		Selectable for each channel: <ul style="list-style-type: none"> <li>• Software trigger</li> <li>• Interrupt requests from peripheral modules or trigger from external interrupt input pins.*1</li> </ul>
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3... > Channel 7 (Channel 0: Highest)
Transfer data	Single data	Bit length: 8, 16, 32 bits
	Block size	Number of data: 1 to 1,024
Transfer mode	Normal transfer mode	<ul style="list-style-type: none"> <li>• One data transfer by one DMA transfer request</li> <li>• Free running function (setting in which total number of data transfers is not specified) settable</li> </ul>
	Repeat transfer mode	<ul style="list-style-type: none"> <li>• One data transfer by one DMA transfer request</li> <li>• Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination.</li> <li>• Maximum settable repeat size: 1,024</li> <li>• Selectable free running function</li> </ul>
	Repeat-block transfer mode	<ul style="list-style-type: none"> <li>• One block data transfer by one DMA transfer request</li> <li>• Maximum settable block size: 1,024</li> <li>• Block transfer can be repeated</li> <li>• Maximum settable repeat size: 64K</li> <li>• Selectable free running function</li> </ul>
	Block transfer mode	<ul style="list-style-type: none"> <li>• One block data transfer by one DMA transfer request</li> <li>• Maximum settable block size: 1,024 data</li> <li>• Selectable free running function</li> </ul>
Selective functions	Extended repeat area function	<ul style="list-style-type: none"> <li>• Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed</li> <li>• Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination</li> </ul>
Processing on DMAC transfer error		<ul style="list-style-type: none"> <li>• When the DMAC transfer error occurs, it is stop the transfer that caused the error channel</li> <li>• Request to clear the register for activation request of DMAC error channel to ICU</li> </ul>
Interrupt (DMAC0n_INT)	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.
	Transfer escape end interrupt	<ul style="list-style-type: none"> <li>• Generated when the repeat size of data transfer is completed.</li> <li>• Generated when the source address extended repeat area overflows.</li> <li>• Generated when the destination address extended repeat area overflows.</li> </ul>
Interrupt (DMA0_TRANS_ERR)	Error response detection interrupt	<ul style="list-style-type: none"> <li>• Generated when the DMAC transfer error occurs</li> </ul>
Event link activation (DMAC0n_INT)		An event link request is generated after each data transfer (for block transfer, after each block is transferred).
Module-stop function		Module-stop state can be set to reduce power consumption.

**Table 16.1 DMAC specifications (2 of 2)**

Item	Description
TrustZone Filter	Security and Privilege can be set for each channels

Note: Security attribution Register of DMAC channel is described in ICU.ICUSARC

Note 1. For details on DMAC activation sources, see [Table 13.4](#) in [section 13, Interrupt Controller Unit \(ICU\)](#).



**Figure 16.1 Block Diagram of DMAC**

## 16.2 Register Descriptions

### 16.2.1 DMACCHSAR : DMA channel Security Attribution Register

Base address: CPSCU = 0x4000\_8000  
CPSCU\_NS = 0x5000\_8000

Offset address: 0x1A0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SADM AC7	SADM AC6	SADM AC5	SADM AC4	SADM AC3	SADM AC2	SADM AC1	SADM AC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SADMAC7 to SADMAC0	Security attributes of output and registers for DMAC channel 0: Secure 1: Non-secure	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

#### SADMACn (n = 0 to 7)

Security attributes registers for DMAC channel.

This bit determines the security attribute of the output as the master for each DMAC channel.

The controlled channel of DMAC registers are shown below.

- CPSCU.DMACCHPAR.PADMACn
- DMAC.DELSR channel n
- DMAC.DMSAR channel n
- DMAC.DMSRR channel n
- DMAC.DMDAR channel n
- DMAC.DMDRR channel n
- DMAC.DMCRA channel n
- DMAC.DMCRB channel n
- DMAC.DMTMD channel n
- DMAC.DMINT channel n
- DMAC.DMAMD channel n
- DMAC.DMOFR channel n
- DMAC.DMCNT channel n
- DMAC.DMREQ channel n
- DMAC.DMSTS channel n
- DMAC.DMSBS channel n
- DMAC.DMDBS channel n
- DMAC.DMBWR channel n



## 16.2.2 DMACCHPAR : DMA channel Privilege Attribution Register

Base address: CPSCU = 0x4000\_8000  
CPSCU\_NS = 0x5000\_8000

Offset address: 0x1F0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	PADM AC7	PADM AC6	PADM AC5	PADM AC4	PADM AC3	PADM AC2	PADM AC1	PADM AC0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	PADMAC7 to PADMAC0	Privilege attributes of outputs and registers for DMAC channel 0: Privileged. 1: Unprivileged.	R/W
31:8	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1

### PADMACn (n = 0 to 7)

Privilege attributes of output and registers for DMAC channel.

This bit determines the Privilege attribute of the output as the master for each DMAC channel. It also determines the privileged attributes of DMAC registers for each channel. (See [section 16.3.14. Channel Privilege](#).)

The controlled the channel of DMAC registers are shown below.

- DMAC.DMSAR channel n
- DMAC.DMSRR channel n
- DMAC.DMDAR channel n
- DMAC.DMDRR channel n
- DMAC.DMCRA channel n
- DMAC.DMCRB channel n
- DMAC.DMTMD channel n
- DMAC.DMINT channel n
- DMAC.DMAMD channel n
- DMAC.DMOFR channel n
- DMAC.DMCNT channel n
- DMAC.DMREQ channel n
- DMAC.DMSTS channel n
- DMAC.DMSBS channel n
- DMAC.DMDBS channel n
- DMAC.DMBWR channel n

### 16.2.3 DMAC SAR : DMAC Controller Security Attribution Register

Base address: CPSCU = 0x4000\_8000  
 CPSCU\_NS = 0x5000\_8000

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMAS TSA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DMASTSA	DMAST Security Attribution 0: Secure 1: Non-secure	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

Note: Only Secure access can write to this register. Both Secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR register.

For DMAC, security attribution is set for each channel. However, this register only sets the DMAST register security attribute.

#### DMASTSA bit (DMAST Security Attribution)

Security attributes of registers for DMAST. Do not write to DMASTSA bit while DMA transfer is enabled or a bus master is writing to the DMA registers.

### 16.2.4 DELSRn : DMAC Event Link Setting Register n (n = 0 to 7)

Base address: DMA0 = 0x4000\_A800  
 DMA0\_NS = 0x5000\_A800

Offset address: 0x080+ 0x04 × n (n = 0 to 7)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IR		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Bit field:	—	—	—	—	—	—	—	DELS[8:0]									—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Bit	Symbol	Function	R/W
8:0	DELS[8:0]	DMAC Event Link Select 0x00: Disable interrupts to the associated DMAC module Others: Event signal number to be linked. For details, see <a href="#">Table 13.4</a> .	R/W <sup>1</sup>
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	IR	DMAC Activation Request Status flag 0: No DMAC activation request occurred. 1: DMAC activation request occurred.	R/W <sup>2</sup>
31:17	—	These bits are read as 0. The write value should be 0.	R/W <sup>1</sup>

Note: P-TYPE-2 See below for access by security attribution.

Note 1. This register must be accessed by half-word or word.

Note 2. Writing 1 to the IR flag is prohibited.

This register has different secure access permission depending on the setting of Trusted Event Route Control Register (TEVTRCR).

If the security attribution is configured as Secure,

- Secure access is allowed.
- Non-secure write access is ignored and Non-secure read access is read as 0, TrustZone access error is generated.

If the security attribution is configured as Non-secure and the trusted event route is disabled,

- Secure write access is ignored and Secure read access is read as 0, TrustZone access error is generated.
- Non-secure access is allowed.

If the security attribution is configured as Non-secure and the trusted event route is enabled,

- Secure access to DELS bit is allowed.
- Non-secure write access to DELS bit is ignored and Non-secure read access to DELS bit is allowed.
- Secure write access to other bits is ignored and Secure read access to other bits is read as 0.
- Non-secure access to other bits is allowed.
- TrustZone access error is not generated.

#### **DELS[8:0] bits (DMAC Event Link Select)**

The DELS[8:0] bits link an event signal to the associated DMAC module. All DELS[8:0] bits must be written to simultaneously. Do not set the same event number in multiple DELSRn registers.

[Setting condition]

- When value is written to these bits.

[Clearing conditions]

- When 0 is written to these bits.
- When DMA transfer is stopped by the access error occurs. See [section 16.5. Processing on DMA Transfer Error](#).

#### **IR flag (DMAC Activation Request Status flag)**

This flag is a status flag for a DMAC activation request.

This flag is associated with the DELS[8:0] bits of this register.

[Setting condition]

- When a DMAC activation request occurs from the associated peripheral module or IRQi pin.

[Clearing conditions]

- When 0 is written to the flag
- When the DMA transfer is started after a DMAC activation request occurs.
- When DMA transfer is stopped by the access error occurs. See [section 16.5. Processing on DMA Transfer Error](#).

Note: IR Flag is automatically cleared after completion of DMA transfer, so do not write "0" except in case of emergency, such as when an abort occurs. DMA transfer operation when "0" is written during DMA transfer can not be guaranteed.

### 16.2.5 DMSAR : DMA Source Address Register

Base address:  $DMAC0n = 0x4000\_A000 + 0x0040 \times n$  (n = 0 to 7)  
 $DMAC0n\_NS = 0x5000\_A000 + 0x0040 \times n$  (n = 0 to 7)

Offset address: 0x00

Bit position: 31 0

Bit field:

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer source start address Setting range is 0x00000000 to 0xFFFFFFFF (4 GB).	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMSAR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer is disabled (DMCNT.DTE = 0).

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bit.

### 16.2.6 DMSRR : DMA Source Reload Address Register

Base address:  $DMAC0n = 0x4000\_A000 + 0x0040 \times n$  (n = 0 to 7)  
 $DMAC0n\_NS = 0x5000\_A000 + 0x0040 \times n$  (n = 0 to 7)

Offset address: 0x20

Bit position: 31 0

Bit field:

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer source reload address 0x0000 0000 to 0xFFFF FFFF (4 GB)	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMSRR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer of the corresponding channel is disabled (DMCNT.DTE = 0).

DMSRR is used to store the start address of the buffer size set in DMSBS during repeat-block transfer mode. In repeat-block transfer mode, DMSAR reloads value of DMSRR after specified transfer finished.

In normal transfer mode, repeat transfer mode and block transfer mode, DMSRR is not used. The setting is invalid.

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bit.

### 16.2.7 DMDAR : DMA Destination Address Register

Base address:  $DMAC0n = 0x4000\_A000 + 0x0040 \times n$  (n = 0 to 7)  
 $DMAC0n\_NS = 0x5000\_A000 + 0x0040 \times n$  (n = 0 to 7)

Offset address: 0x04

Bit position: 31 0

Bit field:

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer destination start address Setting range is 0x0000_0000 to 0xFFFF_FFFF (4 GB).	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMDAR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer of the corresponding channel is disabled (DMCNT.DTE = 0).

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bits.

### 16.2.8 DMDRR : DMA Destination Reload Address Register

Base address: DMAC0n = 0x4000\_A000 + 0x0040 × n (n = 0 to 7)  
 DMAC0n\_NS = 0x5000\_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x24

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the transfer destination reload address Setting range is 0x0000_0000 to 0xFFFF_FFFF (4 GB).	R/W

Note: S-TYPE-3, P-TYPE-3

Set DMDRR while DMAC activation is disabled (DMAST.DMST = 0) or DMA transfer of the corresponding channel is disabled (DMCNT.DTE = 0).

DMDRR is used to store the start address of the buffer size set in DMDBS during repeat-block transfer mode. In repeat-block transfer mode, DMDAR reloads the value of DMDRR after the specified transfer is finished.

In normal transfer mode, repeat transfer mode and block transfer mode, DMDRR is not used. The setting is invalid.

Note: Address alignment in this register must match the Transfer Data Size value selected in the DMTMD.SZ bits.

### 16.2.9 DMCRA : DMA Transfer Count Register

Base address: DMAC0n = 0x4000\_A000 + 0x0040 × n (n = 0 to 7)  
 DMAC0n\_NS = 0x5000\_A000 + 0x0040 × n (n = 0 to 7)

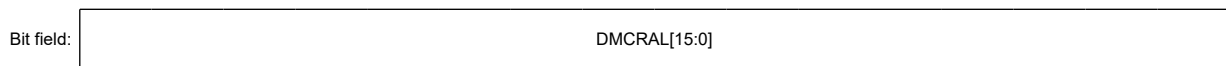
Offset address: 0x08

Bit position: 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	DMCRAL[15:0]	Lower bits of transfer count Specifies the number of transfer operations.	R/W
25:16	DMCRAH[9:0]	Upper bits of transfer count Specifies the number of transfer operations.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Set the same value for DMCRAH and DMCRAL in repeat transfer mode, block transfer mode, and repeat-block transfer mode. Bits 15 to 10 are fixed to 0 in repeat transfer mode, block transfer mode, and repeat-block transfer mode.

**(1) Normal Transfer Mode (DMTMD.MD[1:0] = 00b)**

DMCRAL functions as a 16-bit transfer counter.

The number of transfer operations is one when the setting is 0x0001, and 65,535 when it is 0xFFFF. The value is decremented by one each time data is transferred.

When the setting is 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running function).

Free running function is not selected by DMTMD.TKP bit in normal transfer mode.

DMCRAH is not used in normal transfer mode. Write 0x0000 to DMCRAH.

**(2) Repeat Transfer Mode (DMTMD.MD[1:0] = 01b)**

DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter.

The number of transfer operations is one when the setting is 0x001, 1023 when it is 0x3FF, and 1024 when it is 0x000. In repeat transfer mode, a value in the range of 0x000 to 0x3FF (1 to 1024) can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

**(3) Block Transfer Mode (DMTMD.MD[1:0] = 10b)**

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.

The block size is one when the setting is 0x001, 1023 when it is 0x3FF, and 1024 when it is 0x000. In block transfer mode, a value in the range of 0x000 to 0x3FF can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

**(4) Repeat-Block Transfer Mode (DMTMD.MD[1:0] = 11b)**

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.

The block size is one when the setting is 0x001, 1023 when it is 0x3FF, and 1024 when it is 0x000. In repeat-block transfer mode, a value in the range of 0x000 to 0x3FF can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 0x000, at which the value in DMCRAH is loaded into DMCRAL.

**16.2.10 DMCRB : DMA Block Transfer Count Register**

Base address: DMAC0n = 0x4000\_A000 + 0x0040 × n (n = 0 to 7)  
DMAC0n\_NS = 0x5000\_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMCRBH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DMCRBL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DMCRBL[15:0]	Functions as a number of block, repeat or repeat-block transfer counter. 0x0001 to 0xFFFF (1 to 65535) 0x0000 (65536)	R/W
31:16	DMCRBH[15:0]	Specifies the number of block, repeat or repeat-block transfer operations. 0x0001 to 0xFFFF (1 to 65535) 0x0000 (65536)	R/W

Note: S-TYPE-3, P-TYPE-3

Set the same value for DMCRBH and DMCRBL in repeat transfer mode, block transfer mode and repeat-block transfer mode.

DMCRBH specifies the number of block, repeat and repeat-block transfer operations, and DMCRBL functions as a 16-bit the number of block counter in block, repeat, and repeat-block transfer mode, respectively.

The number of transfer operations is one when the setting is 0x0001, 65535 when it is 0xFFFF, and 65536 when it is 0x0000.

In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred.

In block transfer mode and repeat-block transfer mode, the value is decremented by one when the final data of one block size is transferred.

In normal transfer mode, DMCRB is not used. The setting is invalid.

When DMTMD.TKP is 1 and the final data of one repeat size or one block size is transferred, DMCRBL reloads the value of DMCRBH automatically.

### 16.2.11 DMTMD : DMA Transfer Mode Register

Base address: DMAC0n = 0x4000\_A000 + 0x0040 × n (n = 0 to 7)  
DMAC0n\_NS = 0x5000\_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x10

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MD[1:0]		DTS[1:0]		—	TKP	SZ[1:0]		—	—	—	—	—	—	DCTG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	DCTG[1:0]	Transfer Request Source Select 0 0: Software request 0 1: Hardware request*1 1 0: Setting prohibited 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SZ[1:0]	Transfer Data Size Select 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
10	TKP	Transfer Keeping 0: Transfer is stopped by completion of specified total number of transfer operations. 1: Transfer is not stopped by completion of specified total number of transfer operations (free-running).	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
13:12	DTS[1:0]	Repeat Area Select 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited.	R/W

Bit	Symbol	Function	R/W
15:14	MD[1:0]	Transfer Mode Select 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Repeat-block transfer	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. To select the DMAC activation source, use the DELSRn registers. For details on DMAC activation sources, see [Table 13.4](#) in [section 13, Interrupt Controller Unit \(ICU\)](#).

### DTS[1:0] bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat or block transfer mode. In normal or repeat-block transfer mode, setting these bits is invalid.

### TKP bit (Transfer Keeping)

TKP selects either stopping transfer or keeping transfer by completion of specified total number of transfer operations in repeat, block or repeat-block transfer mode. In normal transfer mode, setting this bit is invalid.

## 16.2.12 DMINT : DMA Interrupt Setting Register

Base address: DMAC0n = 0x4000\_A000 + 0x0040 × n (n = 0 to 7)  
DMAC0n\_NS = 0x5000\_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x13

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable 0: Disables an interrupt request for an extended repeat area overflow on the destination address. 1: Enables an interrupt request for an extended repeat area overflow on the destination address.	R/W
1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable 0: Disables an interrupt request for an extended repeat area overflow on the source address. 1: Enables an interrupt request for an extended repeat area overflow on the source address.	R/W
2	RPTIE	Repeat Size End Interrupt Enable 0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
3	ESIE	Transfer Escape End Interrupt Enable 0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
4	DTIE	Transfer End Interrupt Enable 0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

### DARIE bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while DARIE bit is set to 1, the DMCNT.DTE bit is cleared to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.



When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

When set to repeat-block transfer mode, do not use this bit.

#### SARIE bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while SARIE bit is set to 1, the DMCNT.DTE bit is cleared to 0. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DMCNT.DTE bit of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.

When set to repeat-block transfer mode, do not use this bit.

#### RPTIE bit (Repeat Size End Interrupt Enable)

When RPTIE bit is set to 1 in repeat transfer mode, the DMCNT.DTE bit is cleared to 0 after completion of a 1-repeat size data transfer. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DMCNT.DTE bit is cleared to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the DMSTS.ESIF flag is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DMTMD.DTS[1:0] bits are 10b (= repeat area or block area is not specified).

When set to repeat-block transfer mode, do not use this bit.

#### ESIE bit (Transfer Escape End Interrupt Enable)

ESIE bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.

The transfer escape end interrupt is generated when the DMSTS.ESIF flag is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by clearing this bit or the DMSTS.ESIF flag to 0.

#### DTIE bit (Transfer End Interrupt Enable)

DTIE bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DMSTS.DTIF flag is set to 1 with this bit set to 1. The transfer end interrupt is cleared by clearing this bit or the DMSTS.DTIF flag to 0.

### 16.2.13 DMAMD : DMA Address Mode Register

Base address: DMAC0n = 0x4000\_A000 + 0x0040 × n (n = 0 to 7)  
DMAC0n\_NS = 0x5000\_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x14

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SM[1:0]		SADR	SARA[4:0]				DM[1:0]	DADR	DARA[4:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DARA[4:0]	Destination Address Extended Repeat Area Specifies the extended repeat area on the destination address. For details on the settings, see <a href="#">Table 16.2</a> .	R/W

Bit	Symbol	Function	R/W
5	DADR	Destination Address Update Select After Reload 0: Only reloading. 1: Add index after reloading.	R/W
7:6	DM[1:0]	Destination Address Update Mode 0 0: Destination address is fixed. 0 1: Offset addition. 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
12:8	SARA[4:0]	Source Address Extended Repeat Area Specifies the extended repeat area on the source address. For details on the settings, see <a href="#">Table 16.2</a> .	R/W
13	SADR	Source Address Update Select After Reload 0: Only reloading. 1: Add index after reloading.	R/W
15:14	SM[1:0]	Source Address Update Mode 0 0: Source address is fixed. 0 1: Offset addition. 1 0: Source address is incremented. 1 1: Source address is decremented.	R/W

Note: S-TYPE-3, P-TYPE-3

#### DARA[4:0] bits (Destination Address Extended Repeat Area)

DARA[4:0] bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, and when DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

In repeat-block transfer mode, write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.DARIE bit set to 1. [Table 16.2](#) lists the settings and the corresponding extended repeat areas.

#### DADR bits (Destination Address Update Select After Reload)

In repeat-block transfer mode, this bit specifies the behavior of DMDAR after reloading DMDRR.

When this bit is set to 1, an index value  $((DMDBSH-DMDBSL) \times DataSize)$  is added to DMDAR after reloading DMDRR.

When this bit is set to 0, DMDAR only reloads DMDRR. This behavior is described in [Table 16.13](#).

In normal, repeat or block transfer mode, this bit is ignored.

#### DM[1:0] bits (Destination Address Update Mode)

DM[1:0] bits select the mode of updating the destination address.

When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, or 10b, the destination address is incremented by 1, 2, or 4, respectively.

When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, or 10b, the destination address is decremented by 1, 2, or 4, respectively.

When offset addition is selected, the offset specified by the DMOFR register is added to the address.

#### SARA[4:0] bits (Source Address Extended Repeat Area)

SARA[4:0] bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, and when DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.

In repeat-block transfer mode, write 00000b in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DMINT.SARIE bit set to 1. Table 16.2 lists the settings and the corresponding extended repeat areas.

### SADR bits (Source Address Update Select After Reload)

In repeat-block transfer mode, this bit specifies the behavior of DMSAR after reloading DMSRR.

When this bit is set to 1, an index value  $((\text{DMSBSH}-\text{DMSBSL}) \times \text{DataSize})$  is added to DMSAR after reloading DMSRR.

When this bit is set to 0, DMSAR only reloads DMSRR. This behavior is described in Table 16.12.

In normal, repeat or block transfer mode, this bit is ignored.

### SM[1:0] bits (Source Address Update Mode)

SM[1:0] bits select the mode of updating the source address.

When increment is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, or 10b, the source address is incremented by 1, 2, or 4, respectively.

When decrement is selected and the DMTMD.SZ[1:0] bits are set to 00b, 01b, or 10b, the source address is decremented by 1, 2, or 4, respectively.

When offset addition is selected, the offset specified by the DMOFR register is added to the address.

**Table 16.2 SARA[4:0] or DARA[4:0] settings and corresponding repeat areas (1 of 2)**

SARA[4:0] or DARA[4:0] settings	Extended repeat area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address

**Table 16.2 SARA[4:0] or DARA[4:0] settings and corresponding repeat areas (2 of 2)**

SARA[4:0] or DARA[4:0] settings	Extended repeat area
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

### 16.2.14 DMOFR : DMA Offset Register

Base address:  $DMAC0n = 0x4000\_A000 + 0x0040 \times n$  (n = 0 to 7)  
 $DMAC0n\_NS = 0x5000\_A000 + 0x0040 \times n$  (n = 0 to 7)

Offset address: 0x18

Bit position: 31 0

Bit field:



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination. 0x00000000 to 0x00FFFFFF (0 bytes to (16 M – 1) bytes) 0xFF000000 to 0xFFFFFFFF (–16 Mbytes to –1 byte)	R/W

Note: S-TYPE-3, P-TYPE-3

Write to this register while the DMAC operation is stopped or DMA transfer is disabled (not during data transfer).

Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

In repeat-block transfer mode, the offset is not specified by DMOFR when offset addition is selected, write 0 to DMOFR.

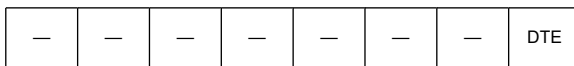
### 16.2.15 DMCNT : DMA Transfer Enable Register

Base address:  $DMAC0n = 0x4000\_A000 + 0x0040 \times n$  (n = 0 to 7)  
 $DMAC0n\_NS = 0x5000\_A000 + 0x0040 \times n$  (n = 0 to 7)

Offset address: 0x1C

Bit position: 7 6 5 4 3 2 1 0

Bit field:



Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	DTE	DMA Transfer Enable 0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

**DTE bit (DMA Transfer Enable)**

When the DMAST.DMST bit is set to 1 (DMAC activation is enabled) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.
- When DMA transfer is stopped by the access error occurs. See [section 16.5. Processing on DMA Transfer Error](#).

If the DTE of the corresponding channel is set to 0 during DMA transfer, no new transfer requests will be accepted after the transfer ends.

**16.2.16 DMREQ : DMA Software Start Register**

Base address: DMAC0n = 0x4000\_A000 + 0x0040 × n (n = 0 to 7)  
DMAC0n\_NS = 0x5000\_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x1D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CLRS	—	—	—	SWREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SWREQ	DMA Software Start 0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	CLRS	DMA Software Start Bit Auto Clear Select 0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

**SWREQ bit (DMA Software Start)**

When 1 is written to SWREQ bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is cleared to 0 if the CLRS bit is set to 0. This bit is not cleared to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DMTMD.DCTG[1:0] bits are set to 00b (DMAC activation source is software).

Setting this bit is invalid when the DMTMD.DCTG[1:0] bits are set to a value other than 00b.

To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

When DMA transfer is stopped by the access error occurs. See [section 16.5. Processing on DMA Transfer Error](#)

### CLRS bit (DMA Software Start Bit Auto Clear Select)

CLRS bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

## 16.2.17 DMSTS : DMA Status Register

Base address: DMAC0n = 0x4000\_A000 + 0x0040 × n (n = 0 to 7)  
DMAC0n\_NS = 0x5000\_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x1E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ACT	—	—	DTIF	—	—	—	ESIF

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	ESIF	Transfer Escape End Interrupt Flag 0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W <sup>1</sup>
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	DTIF	Transfer End Interrupt Flag 0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W <sup>1</sup>
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	ACT	DMAC Active Flag 0: DMAC is in the idle state. 1: DMAC is operating.	R

Note: S-TYPE-3, P-TYPE-3

Note 1. Only 0 can be written to clear the flag.

### ESIF flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the DMINT.RPTIE bit set to 1.
- When 1-block data transfer is completed in block transfer mode with the DMINT.RPTIE bit set to 1.
- When an extended repeat area overflow on the source address occurs while the DMINT.SARIE bit is set to 1 and the DMAMD.SARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer source address).
- When an extended repeat area overflow on the destination address occurs while the DMINT.DARIE bit is set to 1 and the DMAMD.DARA[4:0] bits are set to a value other than 00000b (extended repeat area is specified on the transfer destination address).

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DMCNT.DTE bit.

**DTIF flag (Transfer End Interrupt Flag)**

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of DMCRAL becoming 0 on completion of transfer).
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of DMCRBL becoming 0 on completion of transfer with DMTMD.TKP = 0 or the value of DMCRBL reloading DMCRBH with DMTMD.TKP = 1).
- When the specified number of blocks have been transferred in block transfer mode and repeat-block transfer mode (the value of DMCRBL becoming 0 on completion of transfer with DMTMD.TKP = 0 or the value of DMCRBL reloading DMCRBH with DMTMD.TKP = 1).

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DMCNT.DTE bit.

**ACT flag (DMAC Active Flag)**

This flag indicates whether the DMAC is in the idle or active state.

[Setting condition]

- When the DMAC starts data transfer operation.

[Clearing condition]

- When data transfer in response to one transfer request is completed.

**16.2.18 DMSBS : DMA Source Buffer Size Register**

Base address: DMAC0n = 0x4000\_A000 + 0x0040 × n (n = 0 to 7)  
DMAC0n\_NS = 0x5000\_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMSBSH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DMSBSL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DMSBSL[15:0]	Functions as data transfer counter in repeat-block transfer mode See <a href="#">Table 16.3</a> for available settings.	R/W
31:16	DMSBSH[15:0]	Specifies the repeat-area size in repeat-block transfer mode See <a href="#">Table 16.3</a> for available settings.	R/W

Note: S-TYPE-3, P-TYPE-3

Set the same value for DMSBSH and DMSBSL in repeat-block transfer mode. Write 0x00000000 to DMSBS in normal, repeat and block transfer mode.

DMSBSH specifies buffer size and DMSBSL functions as a 16-bit buffer size counter in repeat-block transfer mode. In repeat-block transfer mode, source repeat area is specified by DMSBSH.

When address update mode is incremented address or decremented address, this register means the numbers of data of whole buffer. When address update mode is offset addition, this register means the numbers of data of an individual buffer. In offset addition, setting DMSBSH and DMSBSL to 0x0000 is prohibited. When final data of one buffer size is transferred,

DMSBSL reloads value of DMSBSH. When address update mode is fixed address, this register is ignored. Table 16.3 shows the setting values of DMA Source Buffer Size Register corresponding to Transfer Data Size in Source Address Update Mode.

**Table 16.3 Available setting for DMSBS register in repeat-block transfer mode**

Source address update mode (DMAMD.SM)	Transfer data size (DMTMD.SZ)	Available setting for DMSBSH and DMSBSL bits
Source address is fixed (SM = 00b)	Don't care	0x0000 (DMSBS is not used)
Offset addition (SM = 01b)	8 bits (SZ = 00b)	0x0001 to 0xFFFF (1 to 65535)
	16 bits (SZ = 01b)	0x0001 to 0x7FFF (1 to 32767)
	32 bits (SZ = 10b)	0x0001 to 0x3FFF (1 to 16383)
Source address is incremented or decremented (SM = 1xb)	Don't care	0x0000 (infinite) 0x0001 to 0xFFFF (1 to 65535)

In normal, repeat and block transfer mode, DMSBS is not used. The setting is invalid.

### 16.2.19 DMDBS : DMA Destination Buffer Size Register

Base address: DMAC0n = 0x4000\_A000 + 0x0040 × n (n = 0 to 7)  
DMAC0n\_NS = 0x5000\_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMDBSH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DMDBSL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DMDBSL[15:0]	Functions as data transfer counter in repeat-block transfer mode. See Table 16.4 for available settings.	R/W
31:16	DMDBSH[15:0]	Specifies the repeat-area size in repeat-block transfer mode. See Table 16.4 for available settings.	R/W

Note: S-TYPE-3, P-TYPE-3

Set the same value for DMDBSH and DMDBSL in repeat-block transfer mode. Write 0x00000000 to DMDBS in normal, repeat and block transfer mode.

DMDBSH specifies buffer size and DMDBSL functions as a 16-bit buffer size counter in repeat-block transfer mode. In repeat-block transfer mode, destination repeat area is specified by DMDBSH.

When address update mode is incremented address or decremented address, this register means the numbers of data of whole buffer. When address update mode is offset addition, this register means the numbers of data of an individual buffer. In offset addition, setting DMDBSH and DMDBSL to 0x0000 is prohibited. When final data of one buffer size is transferred, DMDBSL reloads value of DMDBSH. When address update mode is fixed address, this register is ignored. Table 16.4 shows the setting values of Destination Buffer Size Register corresponding to Transfer Data Size in Destination Address Update Mode.

**Table 16.4 Available setting for DMDBS register in repeat-block transfer mode (1 of 2)**

Destination address update mode (DMAMD.DM)	Transfer data size (DMTMD.SZ)	Available setting for DMDBSH and DMDBSL bits
Destination address is fixed (DM = 00b)	Don't care	0x0000 (DMDBS is not used)



**Table 16.4 Available setting for DMDBS register in repeat-block transfer mode (2 of 2)**

Destination address update mode (DMAMD.DM)	Transfer data size (DMTMD.SZ)	Available setting for DMDBSH and DMDBSL bits
Offset addition (DM = 01b)	8 bits (SZ = 00b)	0x0001 to 0xFFFF (1 to 65535)
	16 bits (SZ = 01b)	0x0001 to 0x7FFF (1 to 32767)
	32 bits (SZ = 10b)	0x0001 to 0x3FFF (1 to 16383)
Destination address is incremented or decremented (DM = 1xb)	Don't care	0x0000 (infinite) 0x0001 to 0xFFFF (1 to 65535)

In normal, repeat and block transfer mode, DMDBS is not used. The setting is invalid.

### 16.2.20 DMBWR : DMA Bufferable Write Enable Register

Base address: DMAC0n = 0x4000\_A000 + 0x0040 × n (n = 0 to 7)  
DMAC0n\_NS = 0x5000\_A000 + 0x0040 × n (n = 0 to 7)

Offset address: 0x30

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	—	BWE
------------	---	---	---	---	---	---	-----

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	BWE	Bufferable Write Enable 0: Disables Bufferable Write 1: Enables Bufferable Write	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

#### BWE bit (Bufferable Write Enable)

BWE bit indicates bufferable write is either enabled or disabled.

If this bit is 1, even if the write access as DMAC is completed, the actual slave write may not have ended.

When this bit is 1, it requests an early response by writing write data to the temporary buffer if possible to the slave. Therefore, even if write access as DMAC is completed, the actual slave write may not be finished. If an error occurs during write access to the target slave, the error may not be detected and the transfer may not be automatically stopped. In such cases, the error response detection interrupt (DMA\_TRANSERR) will not occur.

See [section 14.4.2. Operations When a Bus Error Occurs](#) for slave groups that support bufferable writes and slave groups for which error response detection interrupts cannot be generated.

Some individual slave modules also support bufferable writes, so Refer to the respective module chapters.

[Setting condition]

- When 1 is written to this bit

[Clearing conditions]

- When 0 is written to this bit

### 16.2.21 DMAST : DMA Module Activation Register

Base address: DMA0 = 0x4000\_A800  
DMA0\_NS = 0x5000\_A800

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DMST

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	DMST	DMAC Operation Enable 0: DMAC activation is disabled. 1: DMAC activation is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE2

#### DMST bit (DMAC Operation Enable)

Setting the DMAST.DMST to 1 enables DMAC activation for all channels. When the DMST bit is set to 1 (DMAC activation is enabled), and 1 is written to the DMCNT.DTE bit (DMA transfer is enabled) for multiple channels, all associated channels can be placed in the transfer request ready state at the same time.

When the DMST bit clears to 0 during DMA transfer, DMA transfer is suspended after the current data transfer associated with a single transfer request completes. To resume DMA transfer, set the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit.

[Clearing condition]

- When 0 is written to this bit.

### 16.2.22 DMECHR : DMAC Error Channel Register

Base address: DMA0 = 0x4000\_A800  
DMA0\_NS = 0x5000\_A800

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DMES TA

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit field:	—	—	—	—	—	—	—	DMEC HSAM	—	—	—	—	—	—	—	—	DMECH

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	DMECH	DMAC Error channel Indicates the channel number causing the error 0 0 0: Error occurred on Channel 0 0 0 1: Error occurred on Channel 1 0 1 0: Error occurred on Channel 2 ⋮ 1 1 1: Error occurred on Channel 7	R
7:3	—	These bits are read as 0. The write value should be 0.	R

Bit	Symbol	Function	R/W
8	DMECHSAM	DMAC Error channel Security Attribution Monitor Indicates the security attribution of a channel causing the error 0: secure channel 1: non-secure channel	R
15:9	—	These bits are read as 0. The write value should be 0.	R
16	DMESTA	DMAC Error Status 0: No DMA transfer error occurred 1: DMA transfer error occurred	R/W <sup>1</sup>
31:17	—	These bits are read as 0. The write value should be 0.	R

Note: P-TYPE-2 See below for access by security attribution.

Note 1. Writing to DMESTA depends on the value of DMECHSAM

When reading this register, it can be accessed from both secure and non-secure access.

When writing this register, it depends on DMECHR.DMESA.

- When DMESA = 1, it can be accessed from secure and non-secure access.
- When DMESA = 0, it can be accessed from secure. An error is returned when write access is performed in a non-secure access.

This register is cleared by a reset caused by a transfer error. Please select interrupt (DMA0\_TRANSERR) in BUS.OADCFG.OAD when you want to debug the program.

#### **DMECH[2:0] bit (DMAC Error channel)**

When a transfer error due to DMA transfer occurs, it stores the channel of DMAC that was violated.

When reset was selected in MMPUOAD.OAD and MSAOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs and DMESTA = 0.

[Clearing condition]

- When 1 is written to DMESTA.

#### **DMECHSAM bit (DMAC Error channel Security Attribution Monitor)**

When a transfer error due to DMA transfer occurs, it indicates the security attribution of the violating DMAC channel.

When reset was selected in MMPUOAD.OAD and MSAOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs and DMESTA = 0.

[Clearing condition]

- When 1 is written to DMESTA.

#### **DMESTA bit (DMAC Error Status)**

Indicates whether or not a DMA transfer error occurred.

DMECH, DMECHSAM, DMESTA are cleared by writing 1 to DMESTA. Writing 0 to DMESTA is ignored.

When reset was selected in MMPUOAD.OAD and MSAOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Set condition]

- When the DMAC transfer error occurs.

[Clearing condition]

- When 1 is written to DMESTA.

Note: When DMECHSAM = 1, it can be cleared in the secure state and non-secure state. DMECHSAM = 0, it can not be cleared in the non-secure state.

### 16.3 Operation

#### 16.3.1 Transfer Mode

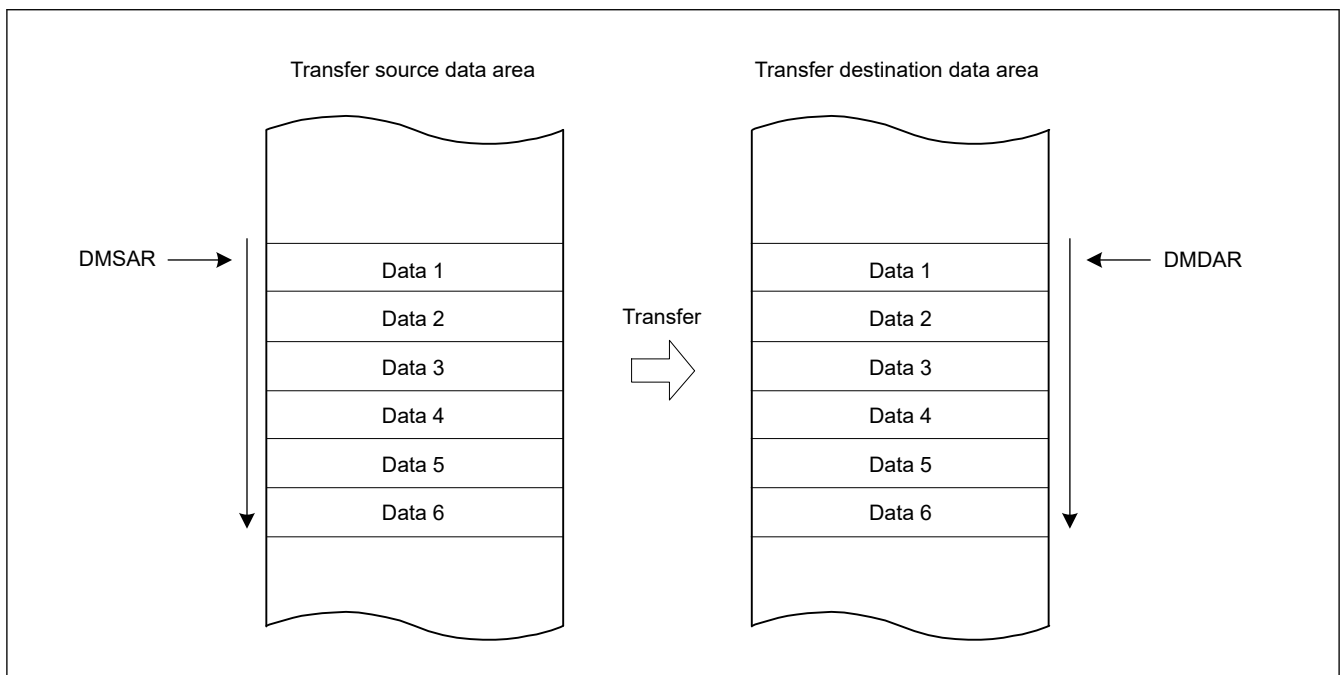
##### 16.3.1.1 Normal Transfer Mode

In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMCRAL register. When these bits are set to 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free-running function). Setting DMCRB register is invalid in normal transfer mode. Except in free-running function, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.

Table 16.5 summarizes the register update operation in normal transfer mode, and Figure 16.2 shows the operation in normal transfer mode.

**Table 16.5 Register update operation in normal transfer mode**

Register	Function	Update operation after completion of a transfer by one transfer request
DMSAR	Transfer source address	Increment/decrement/fixd/offset addition
DMDAR	Transfer destination address	Increment/decrement/fixd/offset addition
DMCRAH	—	Not updated (Not used in normal transfer mode)
DMCRB	—	Not updated (Not used in normal transfer mode)



**Figure 16.2 Operation in normal transfer mode**

##### 16.3.1.2 Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.

A maximum of 1K data can be set as a total repeat transfer size using DMCRA register.

A maximum of 64K can be set as the number of repeat transfer operations using DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of repeat transfer operations) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped, and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the repeat size end interrupt handling.

A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations.

Table 16.6 summarizes the register update operation in repeat transfer mode, and Figure 16.3 shows the operation in repeat transfer mode.

**Table 16.6 Register update operation in repeat transfer mode**

Register	Function	Update operation after completion of a transfer by one transfer request	
		When DMCRAL register is not 1	When DMCRAL register is 1 (Transfer of the last data in repeat size)
DMSAR	Transfer source address	Increment/decrement/fixe d/offset addition	<ul style="list-style-type: none"> <li>• DMTMD.DTS[1:0] = 00b Increment/decrement/fixe d/offset addition</li> <li>• DMTMD.DTS[1:0] = 01b Initial value of DMSAR</li> <li>• DMTMD.DTS[1:0] = 10b Increment/decrement/fixe d/offset addition</li> </ul>
DMDAR	Transfer destination address	Increment/decrement/fixe d/offset addition	<ul style="list-style-type: none"> <li>• DMTMD.DTS[1:0] = 00b Initial value of DMDAR</li> <li>• DMTMD.DTS[1:0] = 01b Increment/decrement/fixe d/offset addition</li> <li>• DMTMD.DTS[1:0] = 10b Increment/decrement/fixe d/offset addition</li> </ul>
DMCRAH	Repeat size	Not updated	Not updated
DMCRAL	Transfer count	Decremente d by one	DMCRAH
DMCRBH	Number of repeat transfer operations	Not updated	Not updated
DMCRBL	Count of repeat transfer operations	Not updated	Decremente d by one

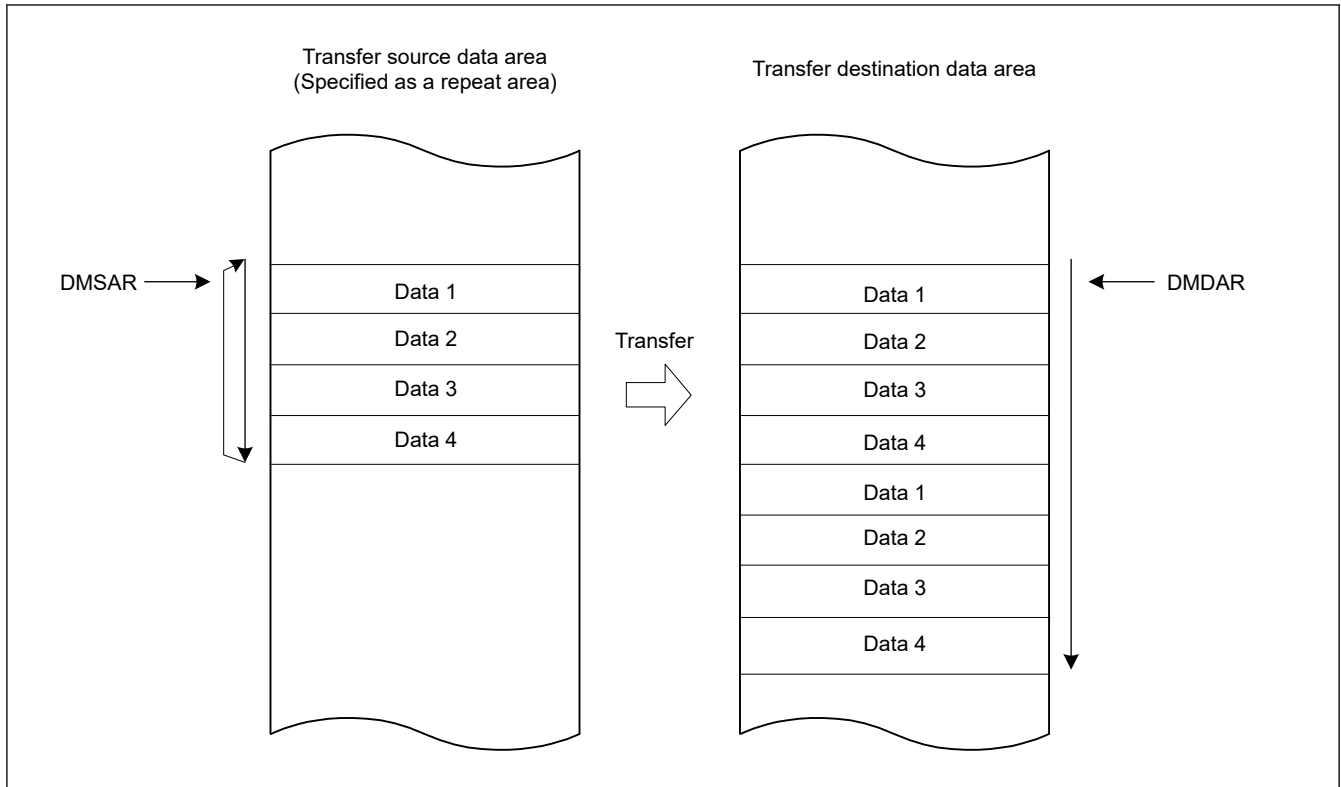


Figure 16.3 Operation in repeat transfer mode

### 16.3.1.3 Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.

A maximum of 1K data can be set as a total block transfer size using DMCRA register.

A maximum of 64K can be set as the number of block transfer operations using DMCRB register; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped, and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the repeat size end interrupt handling.

Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.

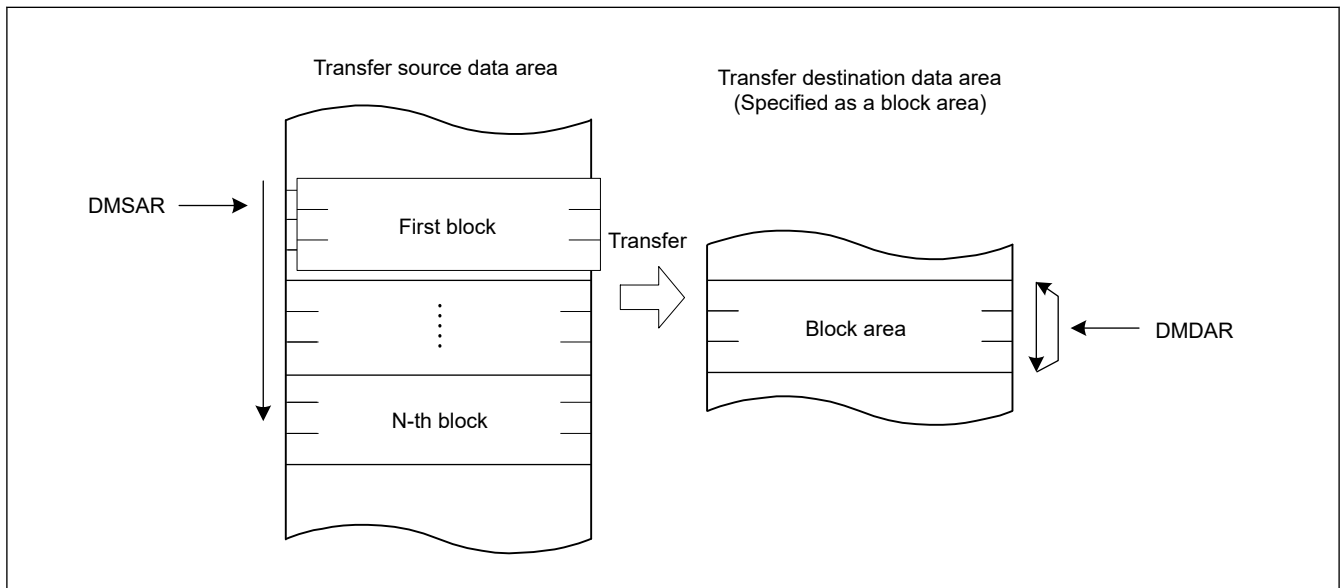
Table 16.7 summarizes the register update operation in block transfer mode, and Figure 16.4 shows the operation in block transfer mode.

Table 16.7 Register update operation in block transfer mode (1 of 2)

Register	Function	Update operation after completion of single-block transfer by one transfer request
DMSAR	Transfer source address	<ul style="list-style-type: none"> <li>DMTMD.DTS[1:0] = 00b Increment/decrement/offset addition</li> <li>DMTMD.DTS[1:0] = 01b Initial value of DMSAR</li> <li>DMTMD.DTS[1:0] = 10b Increment/decrement/offset addition</li> </ul>
DMDAR	Transfer destination address	<ul style="list-style-type: none"> <li>DMTMD.DTS[1:0] = 00b Initial value of DMDAR</li> <li>DMTMD.DTS[1:0] = 01b Increment/decrement/offset addition</li> <li>DMTMD.DTS[1:0] = 10b Increment/decrement/offset addition</li> </ul>

**Table 16.7 Register update operation in block transfer mode (2 of 2)**

Register	Function	Update operation after completion of single-block transfer by one transfer request
DMCRAH	Block size	Not updated
DMCRAL	Transfer count	DMCRAH
DMCRBH	Number of block transfer operations	Not updated
DMCRBL	Count of block transfer operations	Decrement by one

**Figure 16.4 Operation in block transfer mode**

### 16.3.1.4 Repeat-Block Transfer Mode

Repeat-block transfer is the operation mode with the following functions added to the block transfer function.

Repeat function: Added function (ring buffer) to repeat specified address area.

Offset function: Multiple areas with offset can be specified within one block transfer.

The repeat function and the offset function can be used for both the transfer source and the transfer destination of repeat-block transfer.

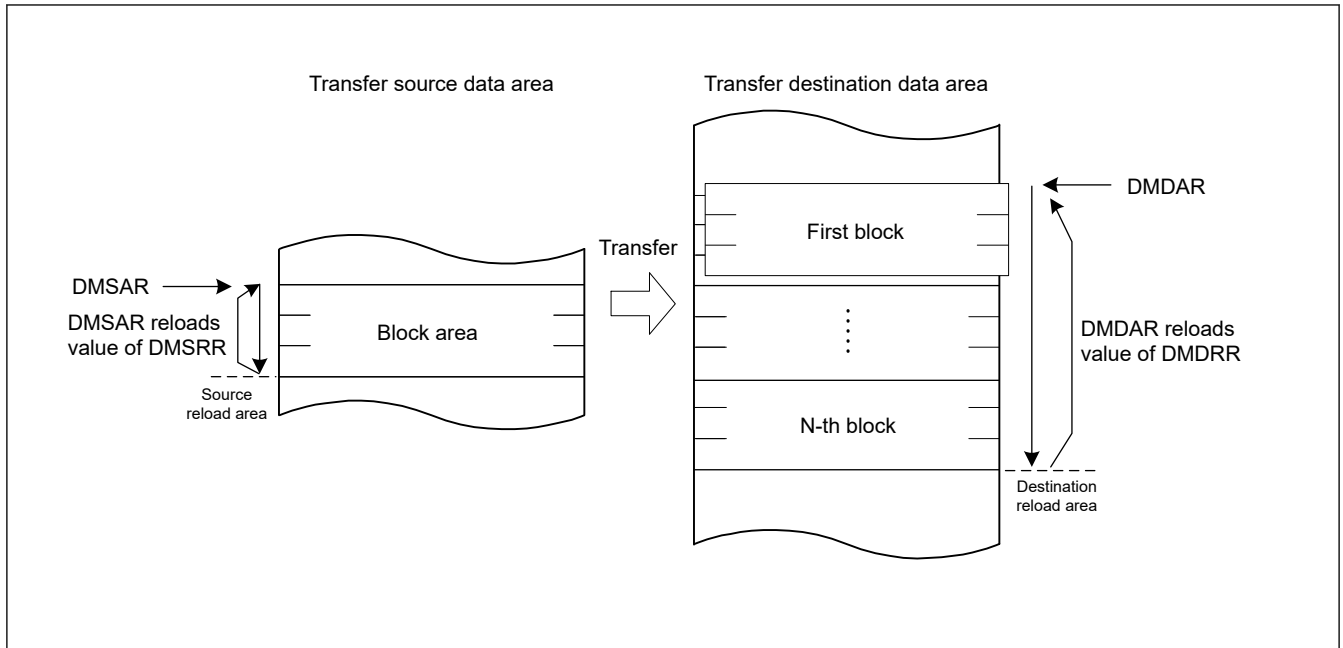
Figure 16.5 shows an example of adding a repeat function to the transfer destination.

Figure 16.6 shows repeat-block transfer with an offset to the transfer destination.

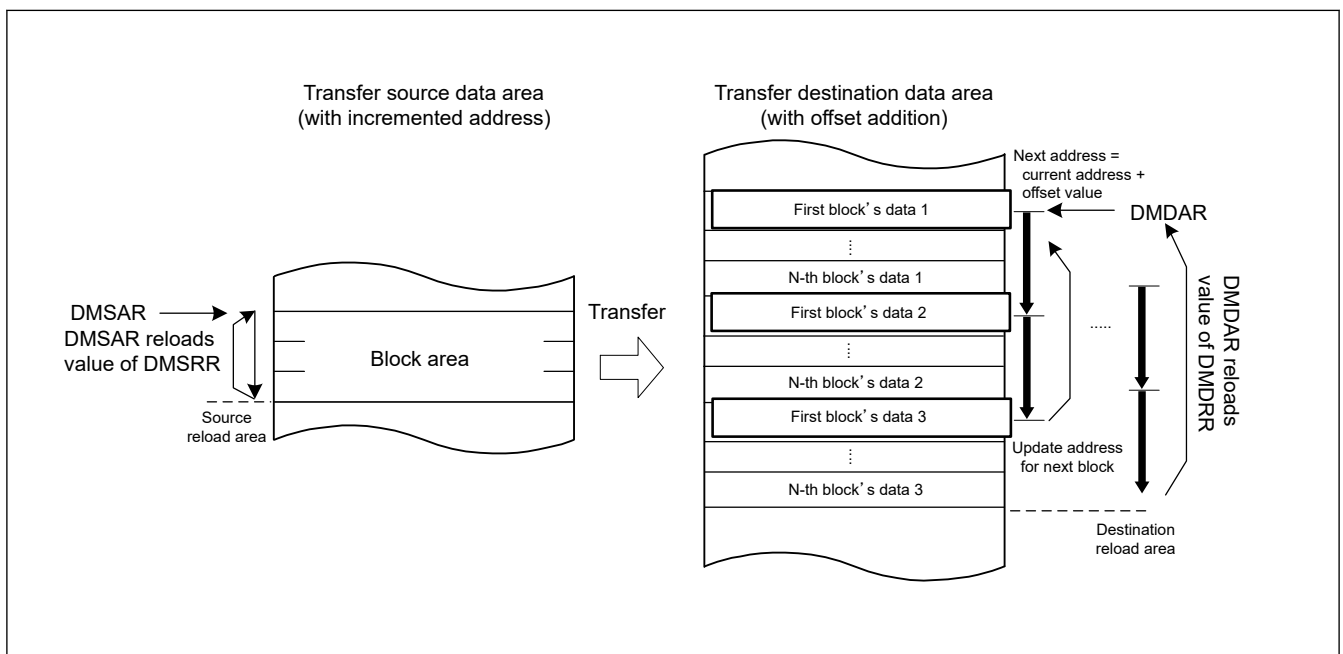
In repeat-block transfer mode, a single block data is transferred by one transfer request.

A maximum of 1K data can be set as a total block transfer size using DMCRA of the DMACn.

A maximum of 64K can be set as the number of block transfer operations using DMCRB of the DMACn; therefore, a maximum of 64M data (1K data × 64K counts of block transfer operations) can be set as a total data transfer size.



**Figure 16.5** Operation in repeat block transfer mode



**Figure 16.6** Operation in repeat-block transfer mode with offset addition

Table 16.8 to Table 16.13 summarize the register update operations in repeat-block transfer mode.

For more information about address update function in repeat-block transfer mode, see [section 16.3.5. Address Update Function in Repeat-Block Transfer Mode](#).

**Table 16.8** Register update operation associated with source area in repeat-block transfer mode (fixed address DMAMD.SM[1:0] = 00b) (1 of 2)

Register	Function	Update operation after single data is transferred		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated



**Table 16.8 Register update operation associated with source area in repeat-block transfer mode (fixed address DMAMD.SM[1:0] = 00b) (2 of 2)**

Register	Function	Update operation after single data is transferred		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSAR	Transfer source address	Not updated	Not updated	Not updated
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decrement by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decrement by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]

**Table 16.9 Register update operation associated with destination area in repeat-block transfer mode (fixed address DMAMD.DM[1:0] = 00b)**

Register	Function	Update operation after single data is transferred		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated
DMDAR	Transfer destination address	Not updated	Not updated	Not updated
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decrement by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decrement by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]

**Table 16.10 Register update operation associated with source area in repeat-block transfer mode (incremented or decremented address DMAMD.SM[1:0] = 10b or 11b) (1 of 2)**

Register	Function	Update operation after single data is transferred					
		DMSBSL[15:0] is not 1			DMSBSL[15:0] is 1		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1			
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated

**Table 16.10 Register update operation associated with source area in repeat-block transfer mode (incremented or decremented address DMAMD.SM[1:0] = 10b or 11b) (2 of 2)**

Register	Function	Update operation after single data is transferred					
		DMSBSL[15:0] is not 1			DMSBSL[15:0] is 1		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1			
DMSAR	Transfer source address when DMTMD.SM[1:0] = 10b	Incremented by Data Size			DMSRR		
	Transfer source address when DMTMD.SM[1:0] = 11b	Decrement by Data Size			DMSRR		
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decrement by 1	DMCRAH[9:0]	DMCRAH[9:0]	Decrement by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMSBSH[15:0]	Source buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMSBSL[15:0]	Count of transfer data in source buffer	Decrement by 1	Decrement by 1	Decrement by 1	DMSBSH	DMSBSH	DMSBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decrement by 1	0	Not updated	Decrement by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]			DMCRBH[15:0]

**Table 16.11 Register update operation associated with destination area in repeat-block transfer mode (incremented or decremented address DMAMD.DM[1:0] = 10b or 11b) (1 of 2)**

Register	Function	Update operation after single data is transferred					
		DMDBSL[15:0] is not 1			DMDBSL[15:0] is 1		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1			
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated

**Table 16.11 Register update operation associated with destination area in repeat-block transfer mode (incremented or decremented address DMAMD.DM[1:0] = 10b or 11b) (2 of 2)**

Register	Function	Update operation after single data is transferred					
		DMDBSL[15:0] is not 1			DMDBSL[15:0] is 1		
		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)		DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)	
DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1		DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1			
DMDAR	Transfer destination address when DMTMD.DM[1:0] = 10b	Incremented by Data Size			DMDRR		
	Transfer destination address when DMTMD.DM[1:0] = 11b	Decrement by Data Size			DMDRR		
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decrement by 1	DMCRAH[9:0]	DMCRAH[9:0]	Decrement by 1	DMCRAH[9:0]	DMCRAH[9:0]
DMDBSH[15:0]	Destination buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMDBSL[15:0]	Count of transfer data in destination buffer	Decrement by 1	Decrement by 1	Decrement by 1	DMDBSH	DMDBSH	DMDBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decrement by 1	0	Not updated	Decrement by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]			

**Table 16.12 Register update operation associated with source area in repeat-block transfer mode (offset addition DMAMD.SM[1:0] = 01b) (1 of 2)**

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMSBSL[15:0] is not 1		DMSBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSRR	Transfer source reload address	Not updated	Not updated	Not updated	Not updated	Not updated

**Table 16.12 Register update operation associated with source area in repeat-block transfer mode (offset addition DMAMD.SM[1:0] = 01b) (2 of 2)**

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMSBSL[15:0] is not 1		DMSBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMSAR	Transfer source address when DMAMD.SADR = 0	Offset addition by DMSBSH	DMSRR		DMSRR	
	Transfer source address when DMAMD.SADR = 1		DMSRR + (DMS-BSH - DMSBSL) × DataSize			
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decrement by 1	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]
DMSBSH[15:0]	Source buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated
DMSBSL[15:0]	Count of transfer data in source buffer	Not updated	Decrement by 1	Decrement by 1	DMSBSH	DMSBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decrement by 1	0	Decrement by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]		

**Table 16.13 Register update operation associated with destination area in repeat-block transfer mode (offset addition DMAMD.DM[1:0] = 01b) (1 of 2)**

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMDBSL[15:0] is not 1		DMDBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDRR	Transfer destination reload address	Not updated	Not updated	Not updated	Not updated	Not updated
DMDAR	Transfer destination address when DMAMD.DADR = 0	Offset addition by DMDBSH	DMDRR		DMDRR	
	Transfer destination address when DMAMD.DADR = 1		DMDRR + (DMDBSH - DMDBSL) × DataSize			
DMCRAH[9:0]	Block size	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRAL[15:0]	Block size count	Decrement by 1	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]	DMCRAH[9:0]

**Table 16.13 Register update operation associated with destination area in repeat-block transfer mode (offset addition DMAMD.DM[1:0] = 01b) (2 of 2)**

Register	Function	DMCRAL[15:0] is not 1	DMCRAL[15:0] is 1 (single block is transferred)			
			DMDBSL[15:0] is not 1		DMDBSL[15:0] is 1	
			DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1	DMCRBL[15:0] is not 1	DMCRBL[15:0] is 1
DMDBSH[15:0]	Destination buffer size (Repeat-size)	Not updated	Not updated	Not updated	Not updated	Not updated
DMDBSL[15:0]	Count of transfer data in destination buffer	Not updated	Decrement by 1	Decrement by 1	DMDBSH	DMDBSH
DMCRBH[15:0]	Number of block transfer operations	Not updated	Not updated	Not updated	Not updated	Not updated
DMCRBL[15:0]	Count of block transfer operations when DMTMD.TKP = 0	Not updated	Decrement by 1	0	Decrement by 1	0
	Count of block transfer operations when DMTMD.TKP = 1			DMCRBH[15:0]		

### 16.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR).

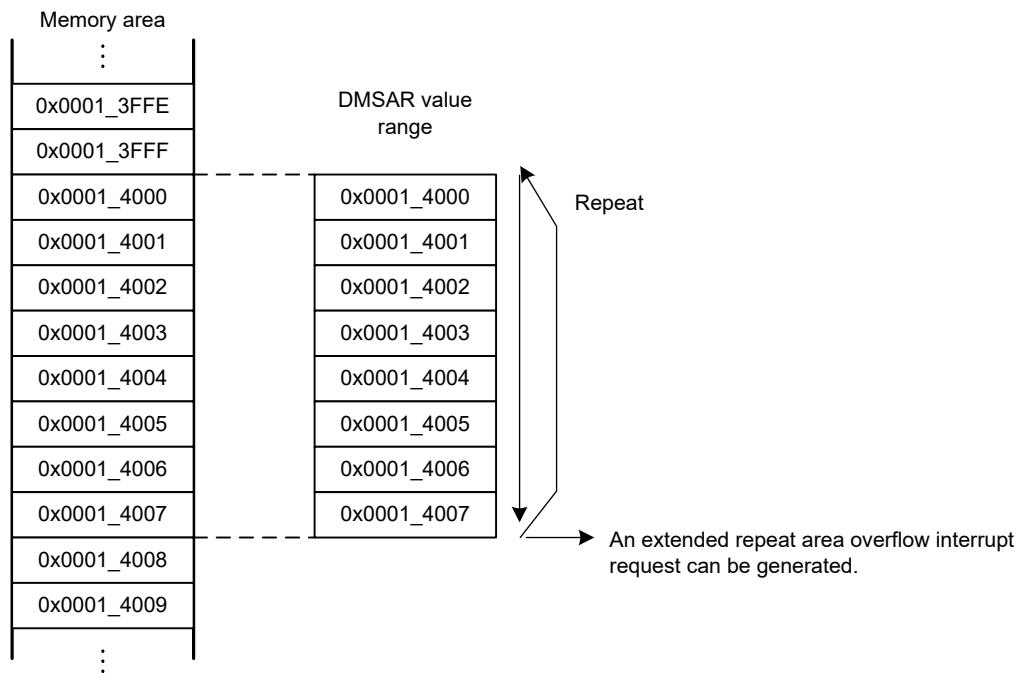
The extended repeat area on the source address is specified by the DMAMD.SARA[4:0] bits. The extended repeat area on the destination address is specified by the DMAMD.DARA[4:0] bits. The size can be specified separately for the source and destination sides.

However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the DMINT.SARIE bit is set to 1, the DMSTS.ESIF flag is set to 1 and the DMCNT.DTE bit is cleared to 0 to stop DMA transfer. At this time, if the DMINT.ESIE bit is set to 1, an interrupt by an extended repeat area overflow is requested. When the DMINT.DARIE bit is set to 1, the destination address register becomes a target to apply the function. DMA transfer can be resumed by writing 1 to the DMCNT.DTE bit in the interrupt handling.

Figure 16.7 shows an example of the extended repeat area operation.

Eight bytes are specified as an extended repeat area by the lower three bits of DMSAR (DMAMD.SARA[4:0] bits = 00011b). The data size is eight bits (DMTMD.SZ[1:0] = 00b).



**Figure 16.7 Example of extended repeat area operation**

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 16.8 shows an example when the extended repeat area function is used in block transfer mode.

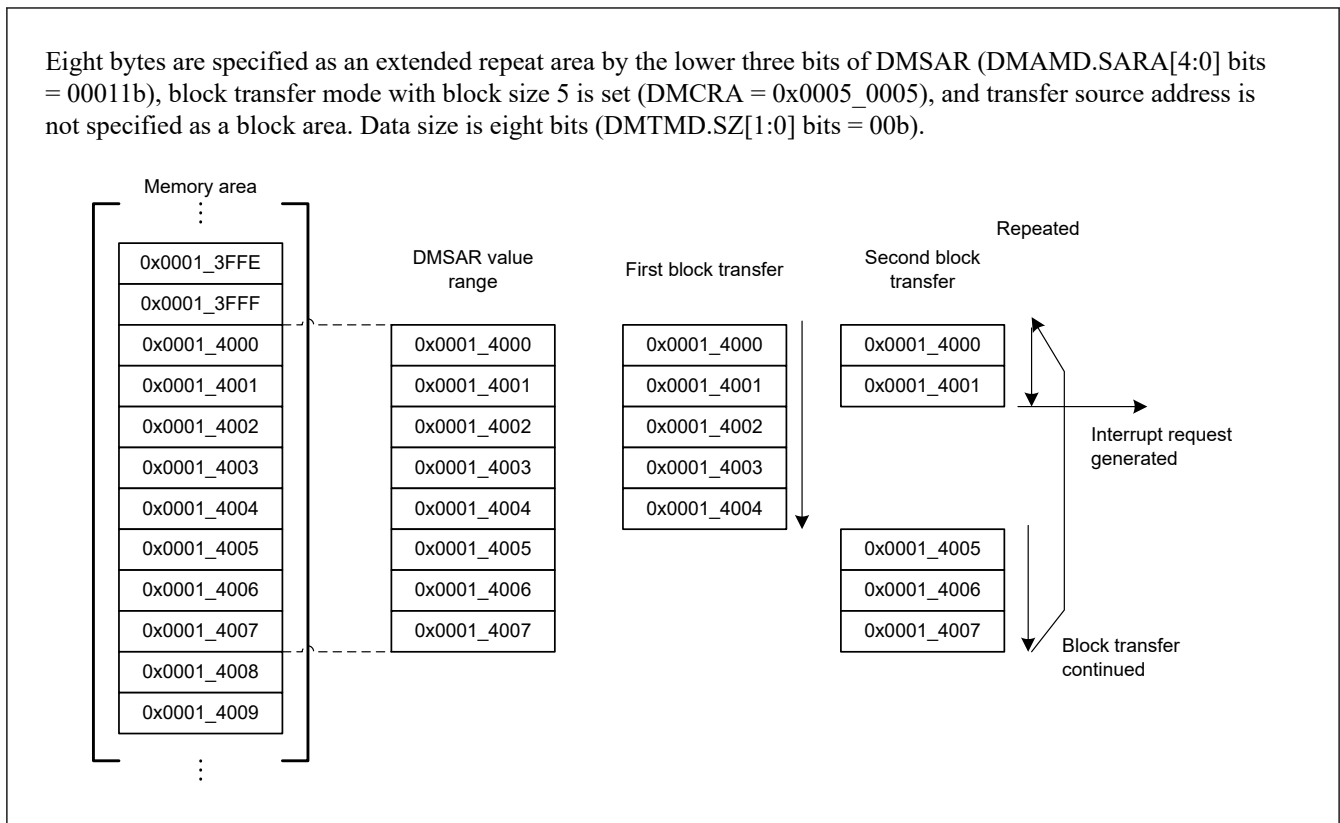


Figure 16.8 Example of extended repeat area function in block transfer mode

### 16.3.3 Free-running Function

The DMAC supports free-running function. This function allows to transfer repeatedly without reconfiguring in interrupt handler.

#### 16.3.3.1 In Normal Transfer Mode

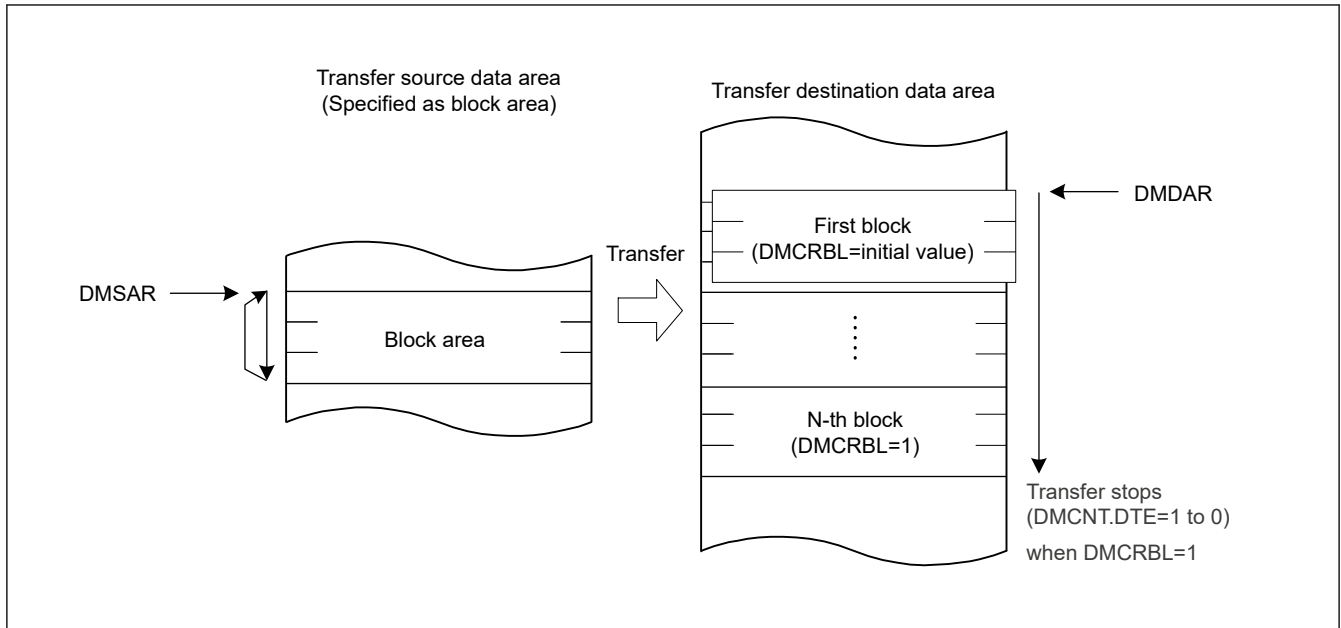
In normal transfer mode, when DMCRA.DMCRAL bits are set to 0x0000, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped.

For more information, see [section 16.3.1.1. Normal Transfer Mode](#).

#### 16.3.3.2 In Other Transfer Modes

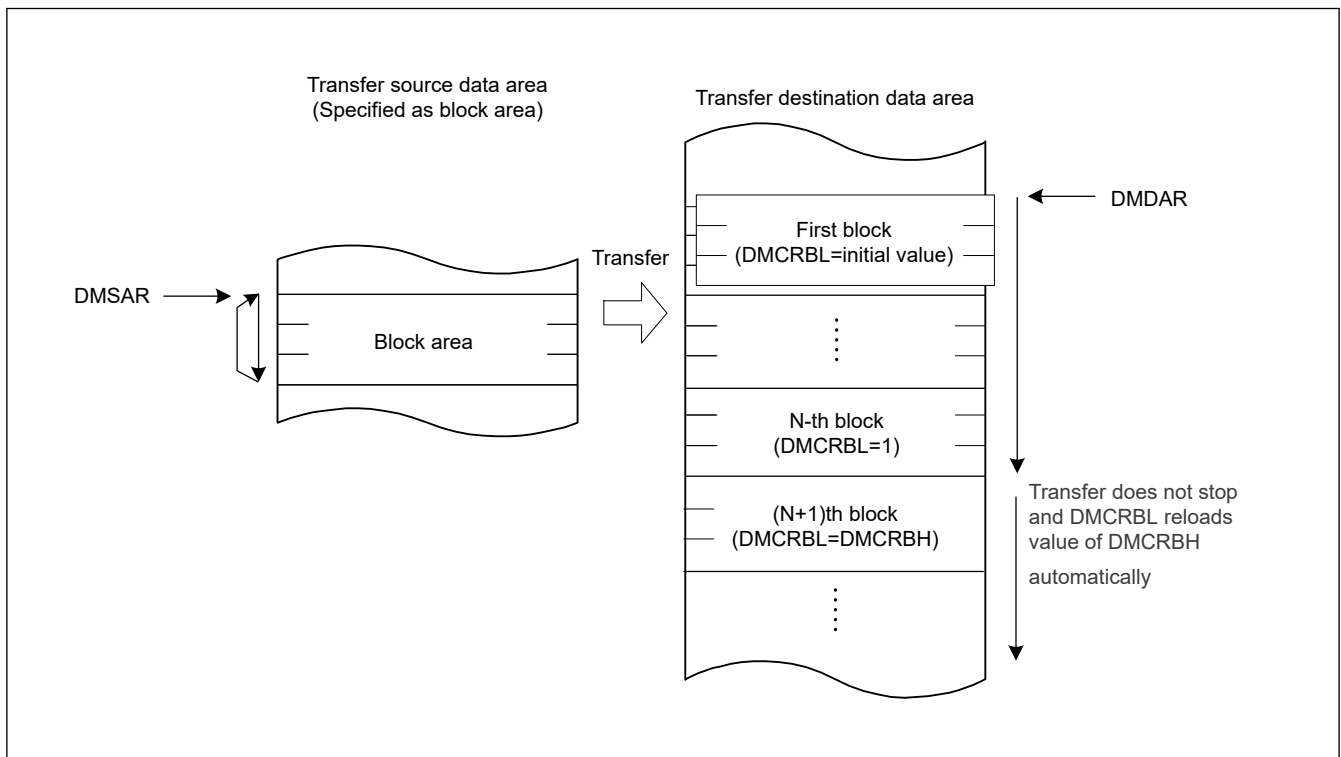
In repeat, block and repeat-block transfer mode, the DMAC supports free-running function using the DMTMD.TKP bit. If the DMTMD.TKP bit is to be set to 1, the transfer is not stopped by completion of specified total number of transfer operations and reloads DMCRBH repeatedly.

Figure 16.9 show an example of block transfer operation without free-running function.



**Figure 16.9** Operation in block transfer mode when DMTMD.TKP bit is set to 0

Figure 16.10 show an example of block transfer operation with free-running function.



**Figure 16.10** Operation in block transfer mode when DMTMD.TKP bit is set to 1

### 16.3.4 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. In normal, repeat and block transfer mode, when the offset addition is selected, the offset specified by the DMA offset register (DMOFR) is added to the address every time the DMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in DMOFR. In this case, the negative value must be 2's complement.



DMSBS or DMDBS are used instead of DMOFR in repeat-block transfer mode. For more information [section 16.3.1.4. Repeat-Block Transfer Mode](#)

Table 16.14 shows the address update method in each address update mode.

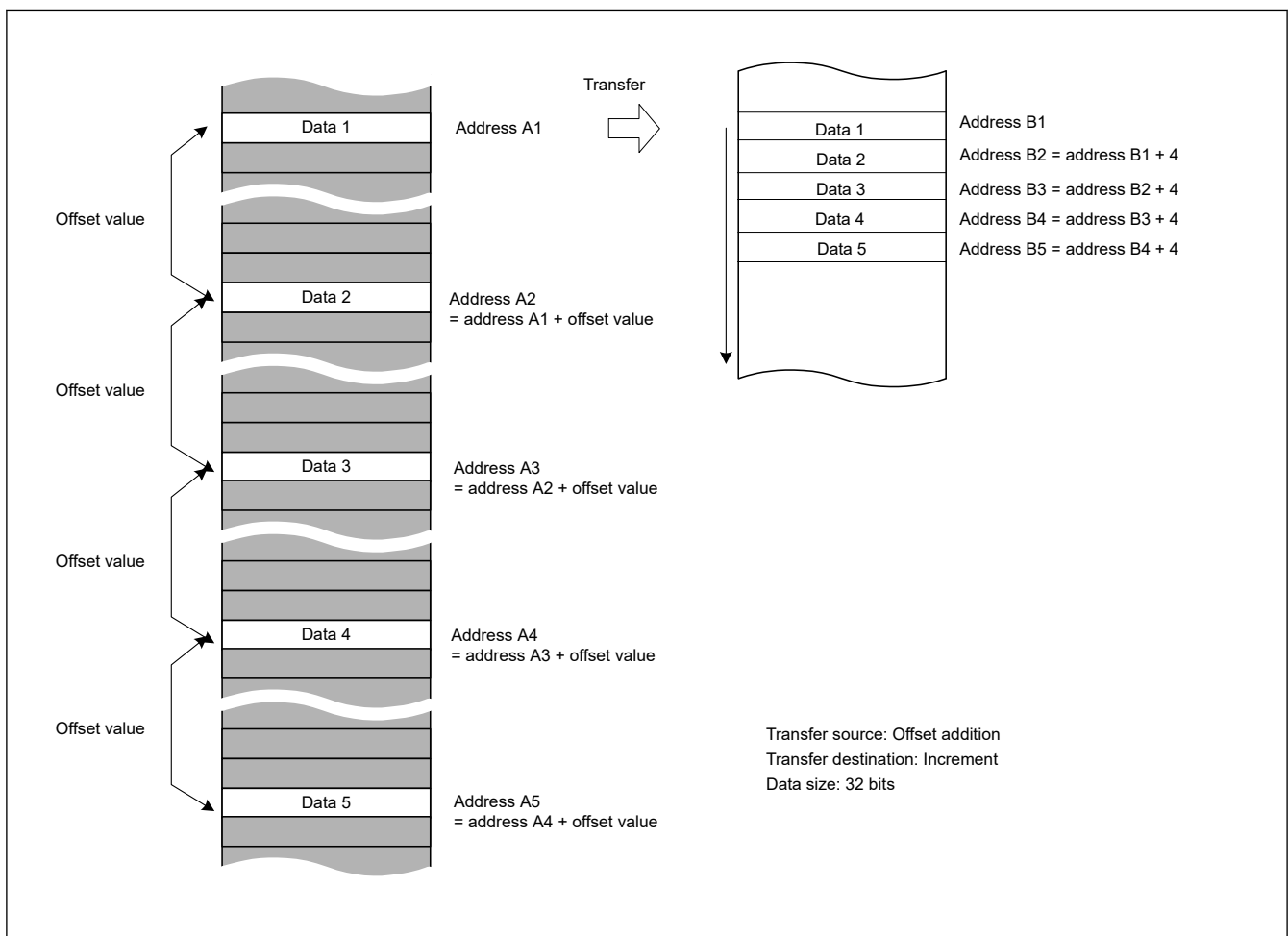
**Table 16.14 Address update method in each address update mode**

Address update mode	Settings of DMAMD.SM[1:0] and DMAMD.DM[1:0] for address update modes	Address update method (for different SZ[1:0] settings in DMTMD)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMOFR <sup>*1</sup>		
Increment	10b	+1	+2	+4
Decrement	11b	-1	-2	-4

Note 1. When setting a negative value in the DMA Offset Register, the value must be in two's complement, obtained by the following formula:  
 two's complement of a negative offset value =  $\sim(\text{offset}) + 1$  ( $\sim$  = bit inversion)

### 16.3.4.1 Basic Transfer Using Offset Addition

Figure 16.11 shows an example of address updating using offset addition.



**Figure 16.11 Example of address updating by offset addition**

Figure 16.11 shows the setting of the following.

- The transfer data is 32 bits long.
- Offset addition is set as the transfer source address update mode.

- Increment is set as the transfer destination address update mode.

The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

### 16.3.4.2 Example of XY Conversion Using Offset Addition

Figure 16.12 shows the XY conversion using offset addition in repeat transfer mode.

Settings are as follows:

- DMAMD.SM — Transfer source address update mode: Offset addition.
- DMAMD.DM — Transfer destination address update mode: Destination address is incremented.
- DMTMD.SZ — Transfer data size select: 32 bits.
- DMTMD.MD — Transfer mode select: Repeat transfer.
- DMTMD.DTS — Repeat area select: The source is specified as the repeat area.
- DMOFR — Offset address: 0x10.
- DMCRA — Repeat size: 0x4.
- DMINT.RPTIE — The repeat size end interrupt is enabled.

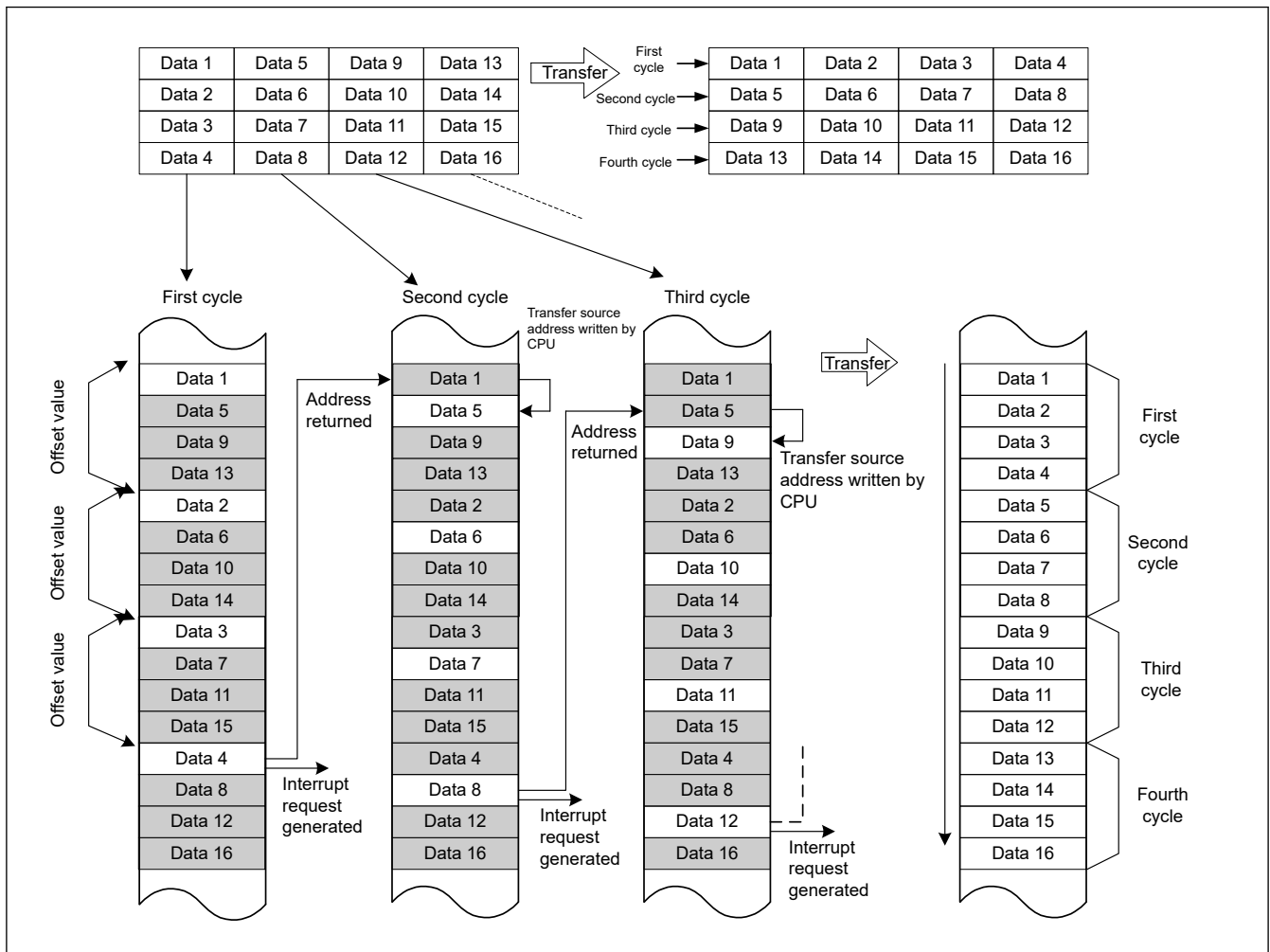


Figure 16.12 XY conversion operation using offset addition in repeat transfer mode

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to continuous transfer destination addresses. When data 4 is transferred:

- The repeat size of data transfer is complete.
- The transfer source address returns to the transfer start address (the address of data 1 on the transfer source).
- A repeat size end interrupt is requested.

During the time this interrupt pauses the transfer, the following operations are performed.

- DMSAR — Rewrite the DMA transfer source address to the address of data 5 (with the above example, the data 1 address + 4).
- DMCNT — Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

[Figure 16.13](#) shows a flowchart of the XY conversion.

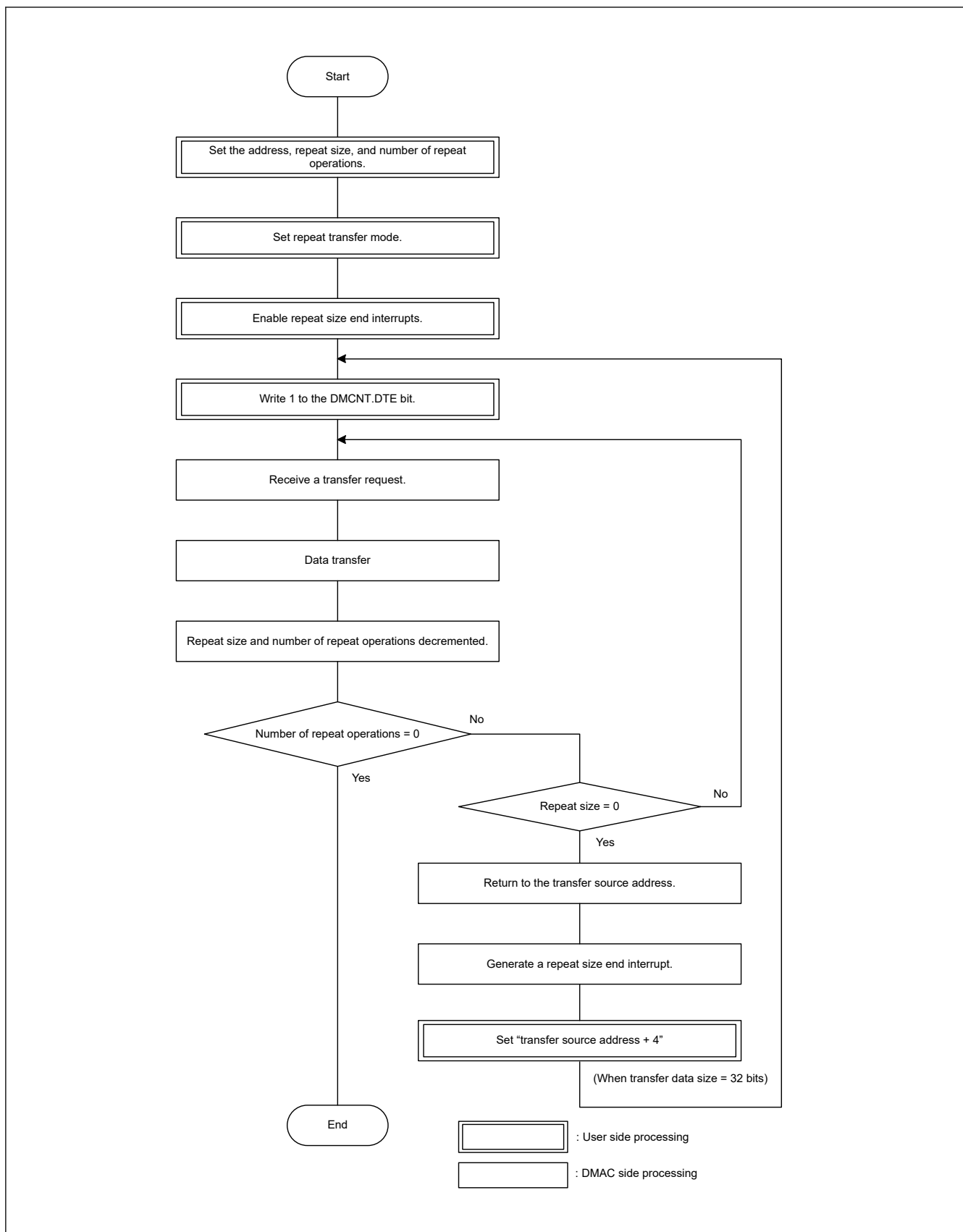


Figure 16.13 XY conversion flowchart using offset addition in repeat transfer mode

### 16.3.5 Address Update Function in Repeat-Block Transfer Mode

Repeat-block transfer mode is an extension of repeat transfer mode and block transfer mode. However, the detailed behavior of the address update is different from these two modes. Here are the details of the address update function in repeat-block transfer mode.

#### 16.3.5.1 Fixed Address Mode

When DMAMD.SM[1:0] is set to 00b, the address update mode of the source is fixed address. And when DMAMD.DM[1:0] is set to 00b, the address update mode of the destination is fixed address.

In fixed address, the address is not updated from the initial value of DMSAR and DMDAR. If the block size (DMCRA) is larger than 1, the same data will be transferred multiple times for one request.

Figure 16.14 shows address update in fixed address.

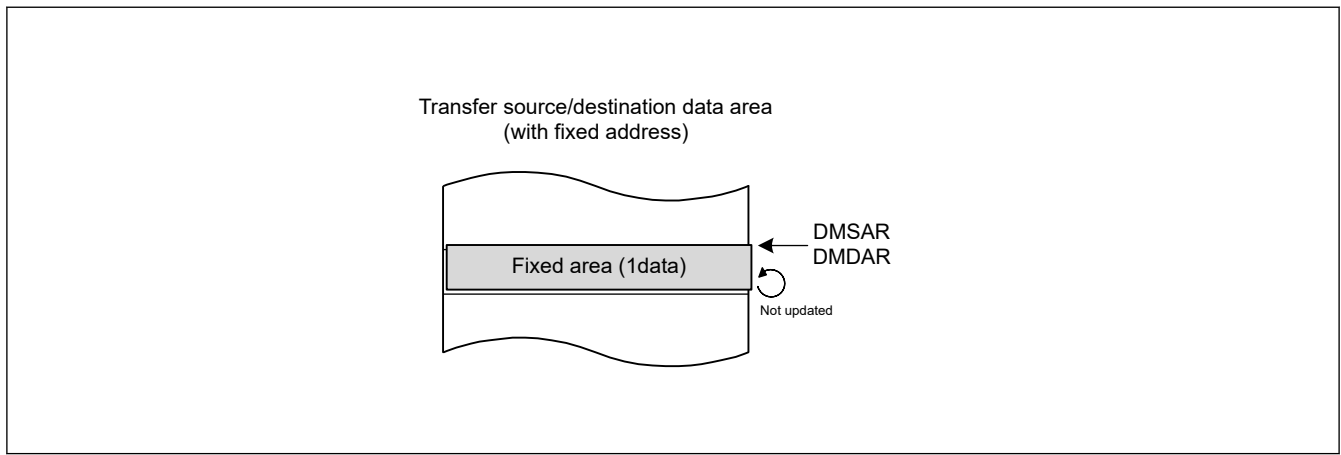


Figure 16.14 Address update in fixed address

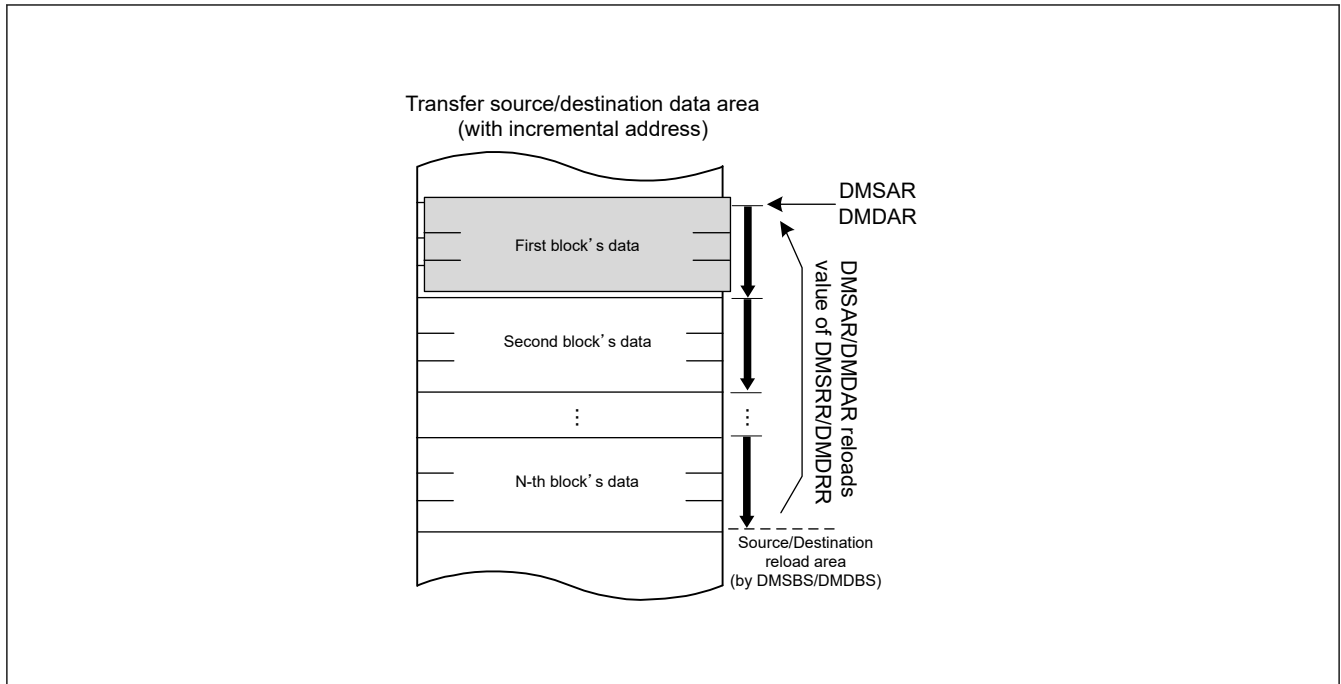
#### 16.3.5.2 Incremental and Decremental Address Mode

When DMAMD.SM[1:0] is set to 10b, the address update mode of the source is incremental address. And when DMAMD.DM[1:0] is set to 10b, the address update mode of the destination is incremental address. When DMAMD.SM[1:0] is set to 11b, the address update mode of the source is decremental address. And when DMAMD.DM[1:0] is set to 11b, the address update mode of the destination is decremental address.

In these update modes, the address is incremented or decremented according to the setting of DMTMD.SZ[1:0].

In these update modes DMSBS and DMDBS indicates a reload area. The unit of DMSBS and DMDBS is "number of data". At the start of transfer, DMSBSL and DMDBSL, which is the lower 16 bits of DMSBS and DMDBS, operates as a down counter and decrements each time one data transfer is performed. When the value becomes 1, DMSAR and DMDAR reloads the value of DMSRR and DMDRR.

Figure 16.15 shows address update in incremental address.



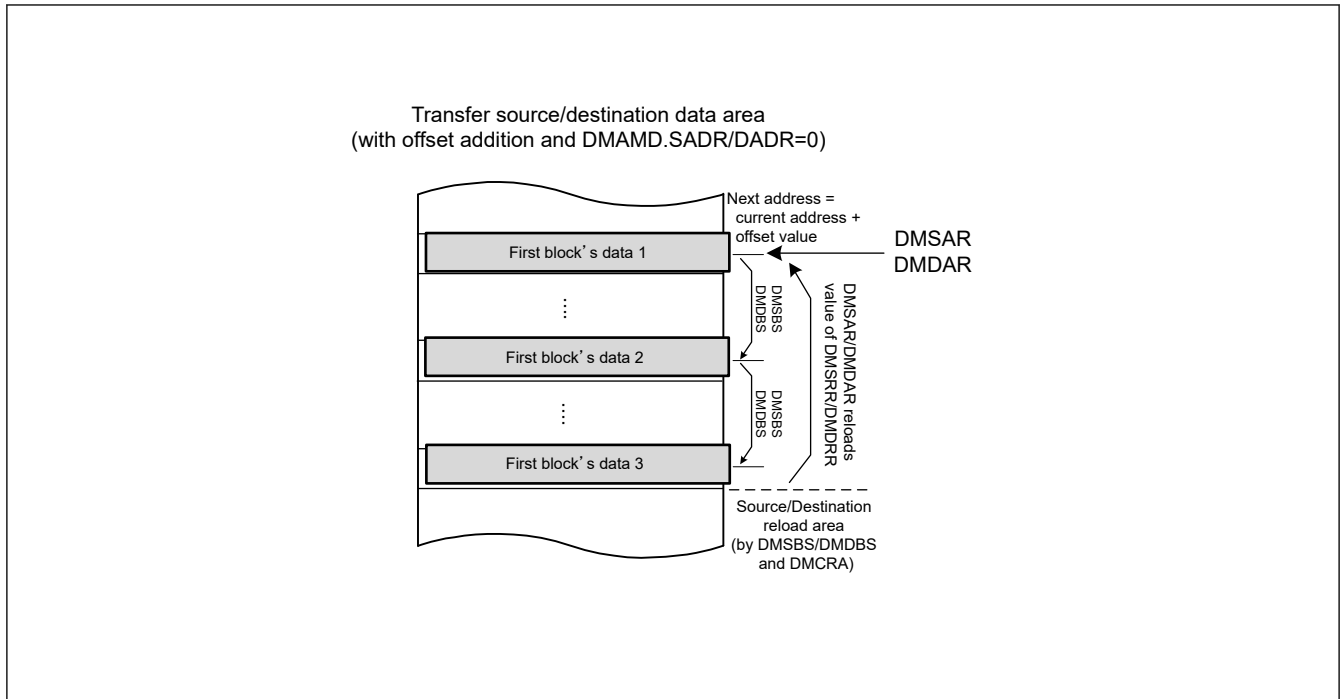
**Figure 16.15 Address update in incremental address**

### 16.3.5.3 Offset Addition Mode

When DMAMD.SM[1:0] is set to 01b, the address update mode of the source is offset addition. And when DMAMD.DM[1:0] is set to 01b, the address update mode of the destination is offset addition.

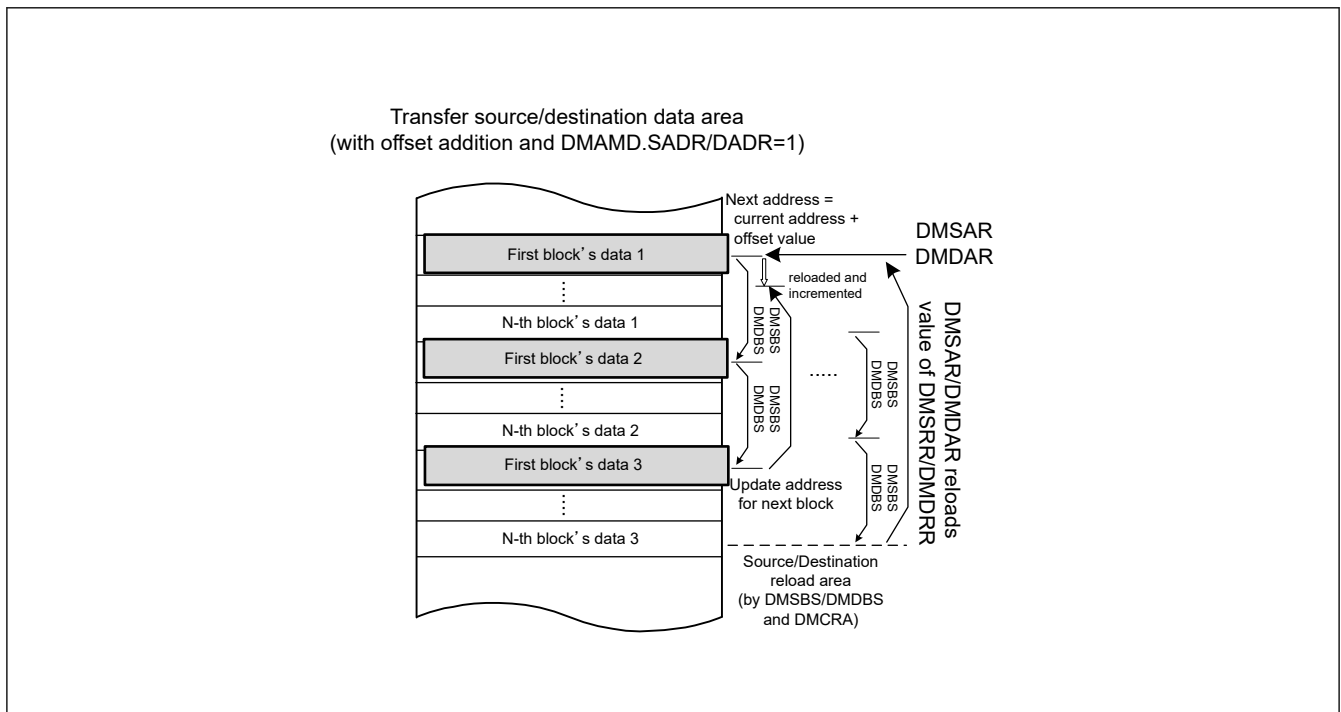
In offset addition, DMSBS and DMDBS indicates reload area and also works as an access offset value. Unlike other transfer modes, DMOFR register is not used in repeat-block transfer mode. In offset addition, the unit of DMSBS and DMDBS is the number of blocks. When the transfer starts, DMCRAL operates as a down counter, DMSAR and DMDAR reloads the value of DMSRR and DMDRR every time one block is transferred. In addition, DMSBSL and DMDBSL, which is the lower 16 bits of DMSBS and DMDBS, also operates as a down counter and decrements every time one block is transferred. When the DMSBS and DMDBS value becomes 1, DMSAR and DMDAR reloads the value of DMSRR and DMDRR.

When DMAMD.SADR and DMAMD.DADR is set to 0, offset addition operation of the same area is repeated. DMDAR only reloads DMDRR. [Figure 16.16](#) shows address update in offset addition with DMAMD.SADR and DMAMD.DADR=0.



**Figure 16.16** Address update in offset addition with DMAMD.SADR and DMAMD.DADR = 0

When DMAMD.SADR and DMAMD.DADR is set to 1, the address is incremented by one data unit after DMSRR and DMDRR is reloaded by DMCRAL=1. In other words, an index value  $((DMDBSH-DMDBSL) \times DataSize)$  is added to DMDAR after DMDRR is reloaded. This behavior is used to implement multiple ring buffers. Figure 16.17 shows address update in offset addition with DMAMD.SADR and DMAMD.DADR=1.



**Figure 16.17** Address update in offset addition with DMAMD.SADR and DMAMD.DADR = 1

### 16.3.6 Example of Using Repeat-Block Transfer Mode

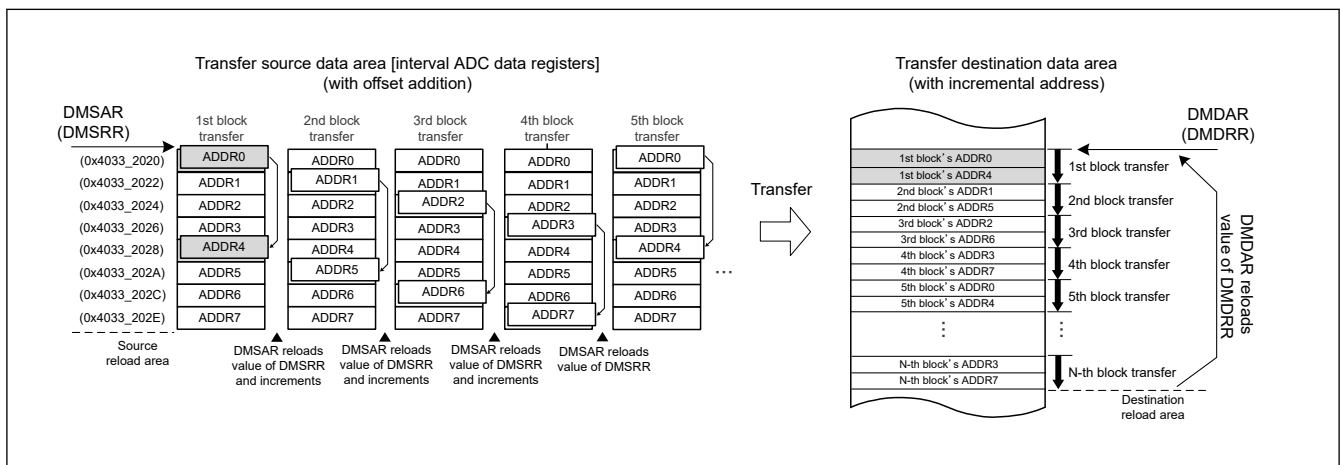
In repeat-block transfer mode, it is possible to realize repeated access to interval data and single or multiple ring buffers by combining the above address update modes. Following sections shows some usage examples.

### 16.3.6.1 Interval Address to Single Ring Buffer

Figure 16.18 shows an example of reading interval ADDRn registers (data register) of ADC12 module and storing it in single ring buffer. It transfers 2 data every 4 halfwords per 1 request. DMSAR is incremented by one data every one request. This can be achieved by setting the transfer source to offset addition and DMAMD.SADR=1, the block size (DMCRA) to 2, and the transfer source offset (DMSBS) to 4. Table 16.15 shows setting of this example.

**Table 16.15 Setting of use case: from interval address to single ring buffer**

Register	Value	Description
DMSAR, DMSRR	0x4033_2020	Initial source address
DMDAR, DMDRR	0x2000_0000	Initial destination address
DMTMD.SZ[1:0]	01b	Data size is halfword
DMAMD.SADR	1	Incremental source address after reloading
DMAMD.SM[1:0]	01b	Source update mode is offset addition
DMAMD.DM[1:0]	10b	Destination update mode is incremental address
DMCRAH, DMCRAL	2	Transfer block size
DMSBSH, DMSBSL	4	Source whole buffer size (unit is 'blocks') and Source access offset (unit is 'data')
DMDBSH, DMDBSL	N × 2 (DMCRA)	Destination buffer size (unit is 'data')



**Figure 16.18 Example of use case: from interval address to single ring buffer**

### 16.3.6.2 Unaligned Ring Buffer to Single Ring Buffer

Figure 16.19 shows an example of reading ADBUFn registers of ADC12 module (conversion result storage ring buffer) incrementally and storing it in single ring buffer. In this example, wrapping occurs because ADBUFn overflows in the fourth scan, but transfer source address of DMAC is also updated accordingly. This can be realized by setting the transfer source to incremental address and setting the DMSBS register to 16 which is the length of ADBUFn. This makes it possible to continue transfer without performing CPU processing using interrupts. Table 16.16 shows setting of this example.

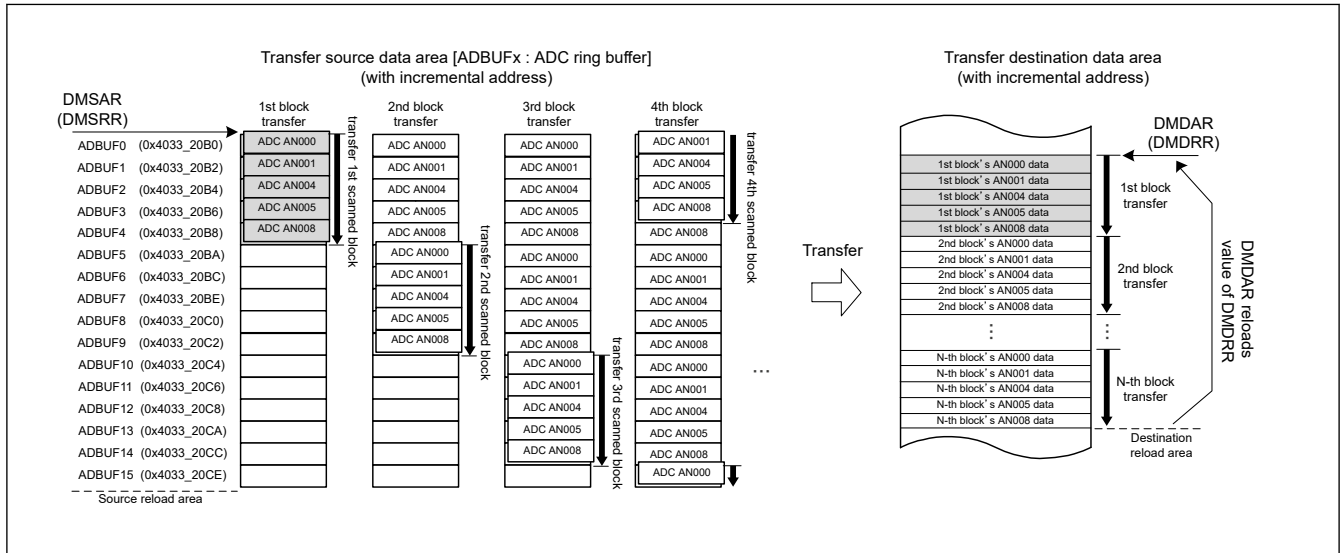
**Table 16.16 Setting of use case: from unaligned ring buffer to single ring buffer (1 of 2)**

Register	Value	Description
DMSAR, DMSRR	0x4033_20B0	Initial source address
DMDAR, DMDRR	0x2000_0000	Initial destination address
DMTMD.SZ[1:0]	01b	Data size is halfword
DMAMD.SM[1:0]	10b	Source update mode is incremental address
DMAMD.DM[1:0]	10b	Destination update mode is incremental address



**Table 16.16** Setting of use case: from unaligned ring buffer to single ring buffer (2 of 2)

Register	Value	Description
DMCRAH, DMCRAL	5	Transfer block size
DMSBSH, DMSBSL	16	Source buffer size (unit is 'data')
DMDBSH, DMDBSL	$N \times 5(\text{DMCRA})$	Destination buffer size (unit is 'data')



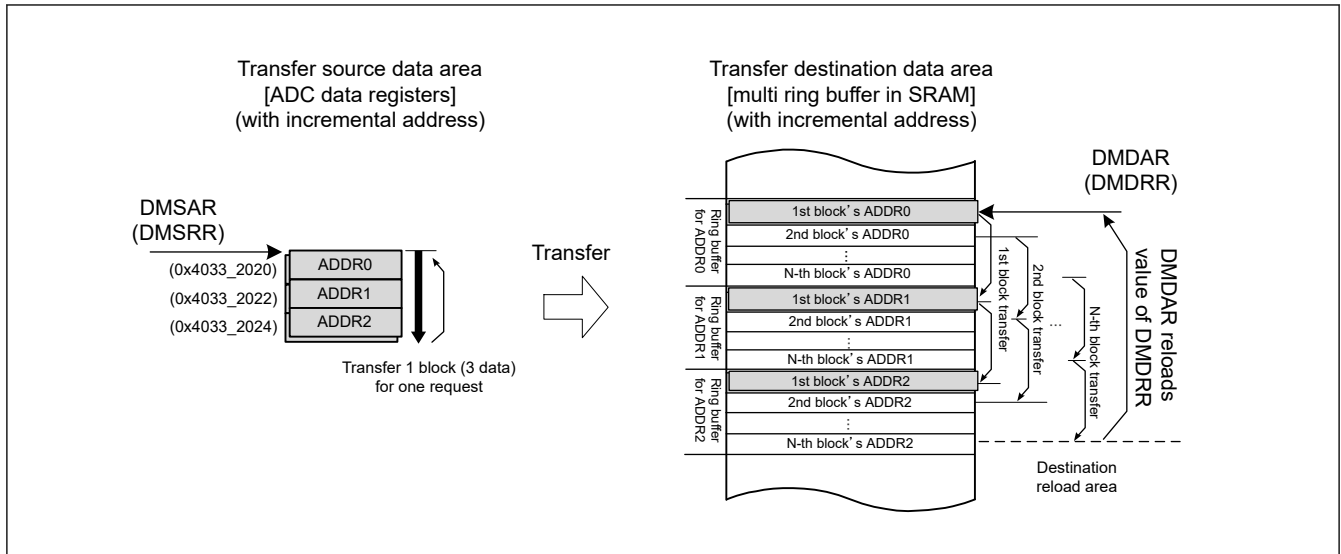
**Figure 16.19** Example of use case: from unaligned ring buffer to single ring buffer

### 16.3.6.3 Single Block to Multi Ring Buffer

Figure 16.20 shows an example of storing the continuous ADDRn registers (data register) of ADC12 module individually in multiple ring buffers. In this example, a ring buffer in which only the first element (ADDR0) in a single block is arranged in transfer order is created at the destination. Also, in the next area, create a ring buffer in which only the second element (ADDR1) is arranged in transfer order. In the following case, create a ring buffer of length N, which is defined by DMDBS. And the number of data elements in the block is 3, which is defined by DMCRA. Table 16.17 shows setting of this example.

**Table 16.17** Setting of use case: from single block to multi ring buffer

Register	Value	Description
DMSAR, DMSRR	0x4033_2020	Initial source address
DMDAR, DMDRR	0x2000_0000	Initial destination address
DMTMD.SZ[1:0]	01b	Data size is halfword
DMAMD.DADR	1	Incremental destination address after reloading
DMAMD.SM[1:0]	10b	Source update mode is incremental address
DMAMD.DM[1:0]	01b	Destination update mode is offset addition
DMCRAH, DMCRAL	3	Transfer block size
DMSBSH, DMSBSL	3	Source buffer size (unit is 'data')
DMDBSH, DMDBSL	N	Destination whole buffer size (unit is 'blocks') and Destination access offset (unit is 'data')



**Figure 16.20 Example of use case: from single block to multi ring buffer**

### 16.3.7 Activation Sources

Software, interrupt requests from the peripheral modules, and external interrupt requests can all be specified as DMAC activation sources. Set the DMTMD.DCTG[1:0] bits to select the activation source.

#### 16.3.7.1 DMAC Activation by Software

When DMA transfer is started by software, follow below procedure.

1. Set the DMTMD.DCTG[1:0] bits to 00b.
2. Set the DMCNT.DTE bit to 1 (DMA transfer is enabled).
3. Set the DMAST.DMST bit set to 1 (DMAC activation enabled).
4. Set the DMREQ.SWREQ bit to 1 (DMA requested).

When the DMAC is activated by software while the DMREQ.CLRS bit is 0, the DMREQ.SWREQ bit is cleared to 0 after data transfer is started in response to a DMA transfer request.

When the DMAC is activated by software while the CLRS bit is 1, the SWREQ bit is not cleared to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

#### 16.3.7.2 DMAC Activation through Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

You can specify interrupt requests from on-chip peripheral modules and external interrupt requests as DMAC activation sources. The activation sources can be selected individually for each channel in DELSRn.DELS[8:0] (n = 0 to 7).

To start DMA transfer through an interrupt request from an on-chip peripheral module or an external interrupt request, follow the procedures as indicated below.

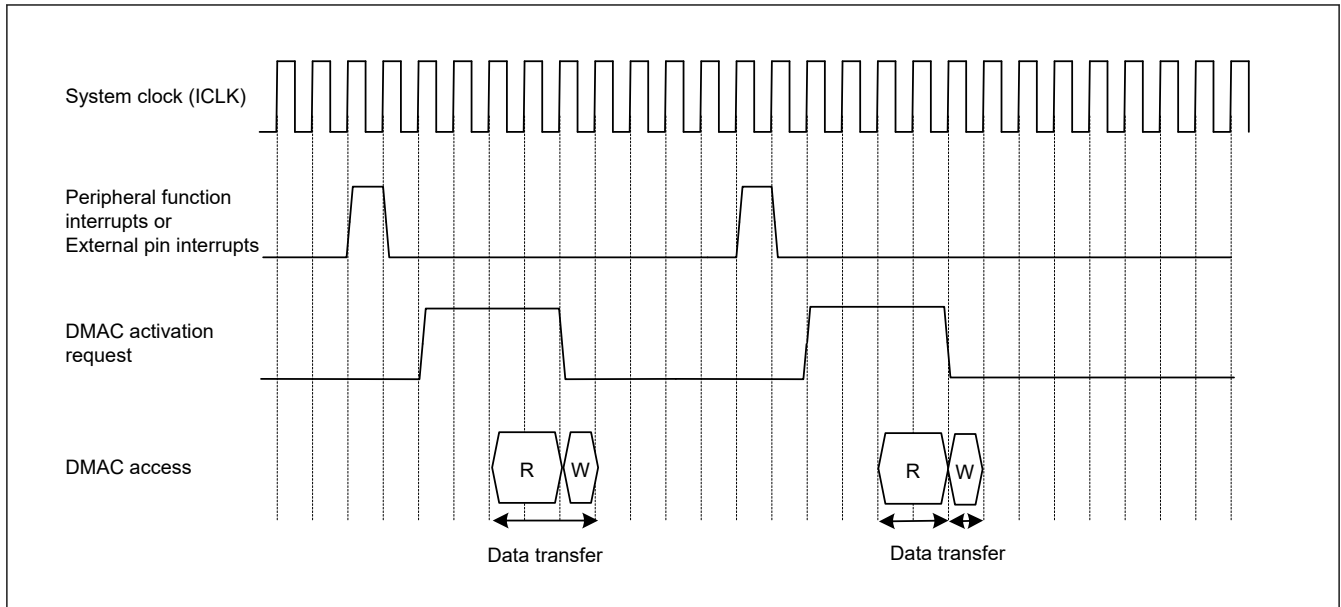
1. Set the DELSR.DELS[8:0] bits to the event number (select the DMAC event link).
2. Set the DMTMD.DCTG[1:0] bits to 01b (interrupts from the peripheral modules and the external interrupt pins).
3. Set the DMCNT.DTE bit to 1 (enable DMA transfer).
4. Set the DMAST.DMST bit set to 1 (DMAC activation enabled)

For interrupt requests specified as DMAC activation sources, see [Table 13.3](#), in [section 13, Interrupt Controller Unit \(ICU\)](#).

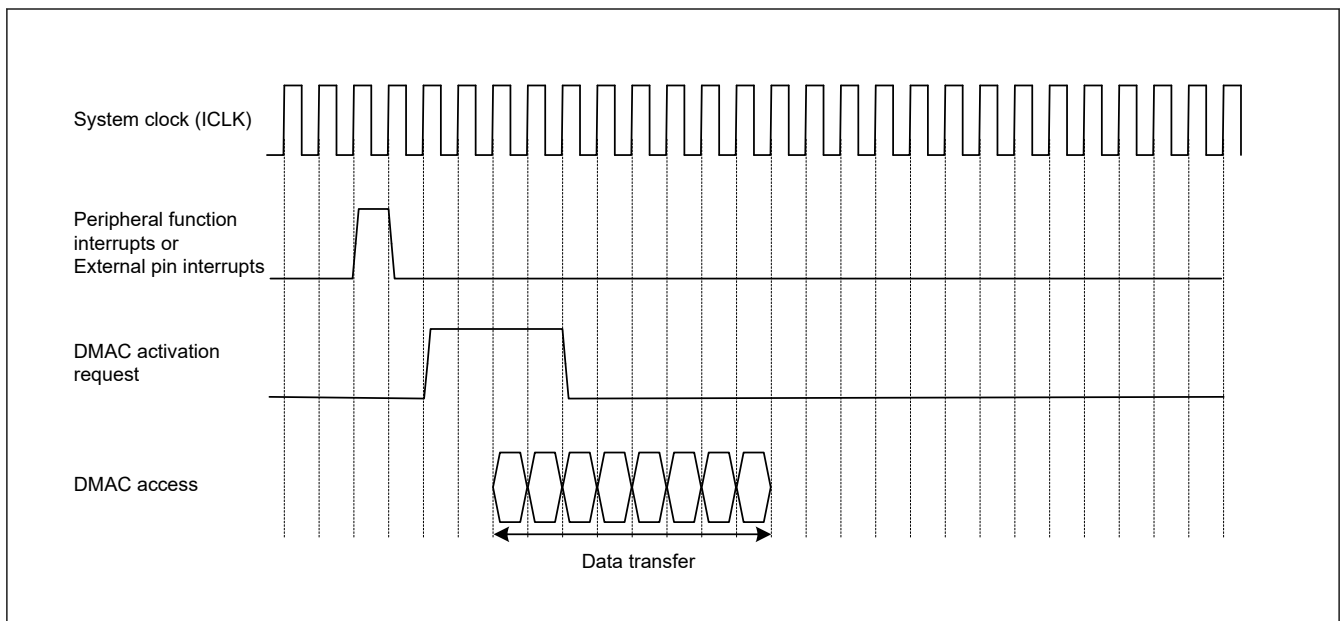
### 16.3.8 Operation Timing

The following timing charts show the minimum number of execution cycles.

[Figure 16.21](#) and [Figure 16.22](#) show DMAC operation timing examples.



**Figure 16.21 DMAC operation timing example 1 with DMAC activation by Interrupt from peripheral module or external interrupt input pin, in normal transfer mode or repeat transfer mode**



**Figure 16.22 DMAC operation timing example 2 with DMAC activation by interrupt from peripheral module or external interrupt input pin, in block transfer mode with block size = 4**

### 16.3.9 Activating the DMAC

Table 16.18 shows the register setting procedure of normal, repeat and block transfer mode and Table 16.19 shows register setting procedure of repeat-block transfer mode.

**Table 16.18 Register Setting Procedure of Normal Transfer Mode, Repeat Transfer Mode and Block Transfer Mode (1 of 2)**

No.	Step Name	Description
1	Disable the peripheral function as the DMAC request source.	To use peripheral function interrupts as DMAC activation sources. Disable the control register for the peripheral function.
2	Disable the IRQn pin as the DMAC request source.	To use external pin interrupts as DMAC activation sources.
3	Set the DELSR.DELS[8:0] bits to 0x00.	Disable the DMAC request.

**Table 16.18 Register Setting Procedure of Normal Transfer Mode, Repeat Transfer Mode and Block Transfer Mode (2 of 2)**

No.	Step Name	Description
4	Clear the DMCNT.DTE bit to 0	Disable DMA transfer.
5	Set the interrupt request as a DMAC request source in the DELSR register.	To use internal peripheral interrupts or external pin interrupts as DMAC activation sources. Enable the interrupt bit for the activation source. Set the DMAC activation source.
6	Set the peripheral module as a DMAC request source	To use peripheral function interrupt as a DMAC activation source. Set the control register for the peripheral function without starting it.
7	Set the IRQn pin function by using the ICU.	To use external pin interrupt as a DMAC activation source. Set the IRQn pin function by using the Interrupt Controller Unit.
8	Set the DMAMD.DM[1:0] bits Set the DMAMD.SM[1:0] bits Set the DMAMD.DARA[4:0] bits Set the DMAMD.SARA[4:0] bits	Set the Transfer destination address update mode bits Set the Transfer source address update mode bits Set the Transfer destination address extended repeat area bits Set the Transfer source address extended repeat area bits
9	Set the DMTMD.DCTG[1:0] bits Set the DMTMD.SZ[1:0] bits Set the DMTMD.DTS[1:0] bits Set the DMTMD.MD[1:0] bits Set the DMTMD.TKP bit	Set the Transfer request select bits Set the Data transfer size bits Set the Repeat area select bits Set the Transfer mode select bits Set the transfer keeping select bit
10	Set the DMSAR register Set the DMDAR register Set the DMCRA register	Set the transfer source start address. Set the transfer destination start address. Set the number of transfer operations.
11	Set the DMCRB register	To use block transfer mode or repeat transfer mode. Set the number of block transfer operations.
12	Set the DMOFR register	To use the address update function with offset. Set the offset value.
13	Set the DMINT.DTIE bit to 1	To use the DMA transfer end interrupts. Enable DMAC transfer end interrupts.
14	Set the DMINT.RPTIE bit Set the DMINT.SARIE bit Set the DMINT.DARIE bit Set the DMINT.ESIE bit to 1	To use the DMA transfer escape end interrupts Set the repeat size end interrupt. Set the transfer source address extended repeat area overflow interrupt. Set the transfer destination address extended repeat area overflow interrupt. Enable the DMA transfer escape end interrupt.
15	Set the DMCNT.DTE bit to 1	Enable DMA transfer.
16	Set the DMAST.DMST bit to 1	Enable DMAC operation. *1 Common settings for DMAC
17	Start the peripheral function as a DMAC request source	To use peripheral function interrupt as a DMAC activation source
18	Enable the IRQn pin as a DMAC request source	To use external pin interrupt as a DMAC activation source
19	End of initial settings	For activation by software On completion of the initial settings, writing 1 to the DMA software start bit (DMREQ.SWREQ) starts DMA transfer.

Note 1. The DMAST.DMST bit setting does not necessarily have to follow the settings for the individual activation sources.

**Table 16.19 Register Setting Procedure of Repeat-Block Transfer Mode (1 of 2)**

No.	Step Name	Description
1	Disable the peripheral function as the DMAC request source.	To use peripheral function interrupts as DMA activation sources. Disable the control register for the peripheral function.
2	Disable the IRQ pin as the DMAC request source.	To use external pin interrupts as DMA activation sources.
3	Set the DELSR.DELS[8:0] bits to 0x00.	Disable the DMAC request.
4	Clear the DMCNT.DTE bit to 0	Disable DMAC transfer.

**Table 16.19 Register Setting Procedure of Repeat-Block Transfer Mode (2 of 2)**

No.	Step Name	Description
5	Set the interrupt request as a DMAC request source in the DELSR register.	To use internal peripheral interrupts or external pin interrupts as DMA activation sources. Enable the interrupt bit for the activation source. Set the DMAC activation source.
6	Set the peripheral module as a DMAC request source	To use peripheral function interrupt as a DMA activation source. Set the control register for the peripheral function without starting it.
7	Set the IRQ pin function by using the Interrupt Controller Unit.	To use external pin interrupt as a DMA activation source. Set the IRQ pin function by using the Interrupt Controller Unit.
8	Set the DMAMD.DM[1:0] bits Set the DMAMD.SM[1:0] bits Set the DMAMD.DADR bit Set the DMAMD.SADR bit	Set the Transfer destination address update mode bits Set the Transfer source address update mode bits Set the Transfer destination address update select after reload Set the Transfer source address update select after reload
9	Set the DMTMD.DCTG[1:0] bits Set the DMTMD.SZ[1:0] bits Set the DMTMD.MD[1:0] bits Set the DMTMD.TKP bit	Set the Transfer request select bits Set the Data transfer size bits Set the Transfer mode to repeat-block transfer mode Set the transfer keeping select bit
10	Set the DMSAR register Set the DMDAR register Set the DMSRR register Set the DMDRR register Set the DMCRA register Set the DMCRB register	Set the transfer source start address Set the transfer destination start address Set the initial value of source start address Set the initial value of destination start address Set the number of transfer operations Set the number of block transfer operations
11	Set the DMSBS register Set the DMDBS register	To use the address update function with incremental, decremental or offset Set the source buffer size and access offset Set the destination buffer size and access offset
12	Set the DMINT.DTIE bit to 1	To use DMA transfer end interrupts. Enable DMAC transfer end interrupts.
13	Set the DMCNT.DTE bit to 1	Enable DMAC transfer
14	Set the DMAST.DMST bit to 1	Enable DMAC operation. *1
15	Start the peripheral function as a DMAC request source	To use peripheral function interrupt as a DMA activation source
16	Enable the IRQ pin as a DMAC request source	To use external pin interrupt as a DMA activation source
17	End of initial settings	For activation by software On completion of the initial settings, writing 1 to the DMA software start bit (DMREQ.SWREQ) starts DMA transfer.

Note 1. The DMAST.DMST bit setting does not necessarily have to follow the settings for the individual activation sources.

### 16.3.10 Starting DMA Transfer

To enable the DMA transfer, set the DMCNT.DTE bit to 1 (enable the DMA transfer), and then set the DMAST.DMST bit to 1 (enable the DMAC activation).

New activation requests are not accepted during the transfer of another DMAC channel or DTC. When the preceding transfer is complete, channel arbitration selects the DMA transfer request of the highest priority channel, and the DMA transfer of that channel starts. When the DMA transfer starts, the DMSTS.ACT flag is set to 1 (the DMAC is in the active state).

### 16.3.11 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMSBS, DMDBS, DMCNT, and DMSTS.

#### DMA Source Address Register (DMSAR)

When data has been transferred in response to one transfer request, the contents of DMSAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, see [Table 16.5](#) to [Table 16.13](#).

#### **DMA Destination Address Register (DMDAR)**

When data has been transferred in response to one transfer request, the contents of DMDAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, see [Table 16.5](#) to [Table 16.13](#).

#### **DMA Transfer Count Register (DMCRA)**

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see [Table 16.5](#) to [Table 16.13](#).

#### **DMA Block Transfer Count Register (DMCRB)**

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see [Table 16.5](#) to [Table 16.13](#).

#### **DMA Source Buffer Size Register (DMSBS)**

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see [Table 16.8](#) to [Table 16.13](#).

#### **DMA Destination Buffer Size Register (DMDBS)**

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, see [Table 16.8](#) to [Table 16.13](#).

#### **DMA Transfer Enable Bit (DMCNT.DTE)**

Although the DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically cleared to 0 by the DMAC according to the DMA transfer state.

The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt
- When DMA transfer error occurs

Writing to the registers for the channels when the corresponding DMCNT.DTE bit is set to 1 is prohibited (except for DMCNT ). In this case, writing must be performed after the bit is cleared to 0.

#### **DMAC Active Flag (DMSTS.ACT)**

The DMSTS.ACT flag indicates whether the DMACn is in the idle or active state.

This flag is set to 1 when the DMAC starts data transfer, and is cleared to 0 when data transfer in response to one transfer request is completed.

Even when DMA transfer is stopped by writing 0 to the DMCNT.DTE bit during DMA transfer, this flag remains 1 until DMA transfer is completed.

#### **Transfer End Interrupt Flag (DMSTS.DTIF)**

The DMSTS.DTIF flag is set to 1 after DMA transfer of the total transfer size of data is completed.

When both this flag and the DMINT.DTIE bit are set to 1, a transfer end interrupt is requested.

This flag is set to 1 when the DMA transfer bus cycle is completed and the DMSTS.ACT flag is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DMCNT.DTE bit is set to 1 during the interrupt handling.

### Transfer Escape End Interrupt Flag (DMSTS.ESIF)

The DMSTS.ESIF flag is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the DMINT.ESIE bit are set to 1, a transfer escape end interrupt is requested.

This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the DMSTS.ACT flag is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DMCNT.DTE bit is set to 1 during an interrupt handling.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set.

For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

## 16.3.12 Channel Priority

When multiple DMA transfer requests are present, the DMAC determines the priority of channels that have DMA transfer requests.

- The channel priority is fixed as follows: Channel 0 > Channel 1 > Channel 2 > Channel 3... > Channel 7 (Channel 0: Highest).

When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

## 16.3.13 Channel Security

The secure attribute can be set with CPSCU.DMACCHSAR.SADMACn for each DMAC channel.

When the CPSCU.DMACCHSAR.SADMACn bit is 0.

- When the corresponding channel transfers, it behaves as a secure master.
- The register of corresponding channel has a secure attribute.

When the CPSCU.DMACCHSAR.SADMACn bit is 1.

- When the corresponding channel transfers, it behaves as a non-secure master.
- The register of corresponding channel has a non-secure attribute.

Refer to the Security chapter and BUS chapter for areas accessible to secure and non-secure masters.

Do not change to the CPSCU.DMACCHSAR while DMA transfer.

## 16.3.14 Channel Privilege

The privileged attribute can be set with CPSCU.DMACCHPAR.DMACCHPARn for each DMAC channel n.

When the CPSCU.DMACCHPAR.DMACCHPARn bit is 0.

- When the corresponding channel transfers, it behaves as a privileged master.
- The registers of channel are protected from a unprivileged access.

When the CPSCU.DMACCHPAR.DMACCHPARn bit is 1.

- The transfer of DMAC channel is unprivileged access for both read and write.
- The registers of channel are unprivileged attributes.

Do not change to the CPSCU.DMACCHPAR while DMA transfer.

## 16.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DMCNT.DTE bit and the DMSTS.ACT flag are changed from 1 to 0, indicating that DMA transfer has ended.

### 16.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

#### (1) In Normal Transfer Mode (DMTMD.MD[1:0] = 00b)

When the value of DMCRAL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

#### (2) In Repeat Transfer Mode (DMTMD.MD[1:0] = 01b)

When the value of DMCRL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

If the DMTMD.TKP bit is 1 (in free-running function), the DMSTS.DTIF bit is set to 1, but the DMCNT.DTE bit is not cleared to 0.

#### (3) In Block Transfer Mode (DMTMD.MD[1:0] = 10b)

When the value of DMCRL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set.

For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

If the DMTMD.TKP bit is 1 (in free-running function), the DMSTS.DTIF bit is set to 1, but the DMCNT.DTE bit is not cleared to 0.

#### (4) In Repeat-Block Transfer Mode (DMTMD.MD[1:0] = 11b)

When the value of DMCRL changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DMCNT.DTE bit is cleared to 0 and the DMSTS.DTIF flag is set to 1 at the same time. If the DMINT.DTIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

If the DMTMD.TKP bit is 1 (in free-running function), the DMSTS.DTIF bit is set to 1, but the DMCNT.DTE bit is not cleared to 0.

### 16.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the DMINT.RPTIE bit is set to 1. When the interrupt is requested to complete DMA transfer, the DMCNT.DTE bit is cleared to 0 and the DMSTS.ESIF flag is set to 1 even if the DMTMD.TKP bit is 1 (in free-running function). If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DMCNT.DTE bit.

A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size data is completed.

Repeat size end interrupt cannot be requested in repeat-block transfer mode.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

### 16.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the DMINT.SARIE or DMINT.DARIE bit is set to 1 even if the DMTMD.TKP bit is 1 (in free-running function), an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DMCNT.DTE bit is cleared to 0, and the ESIF flag in DMSTS is set to 1. If the DMINT.ESIE bit is 1 at this time, an interrupt request is issued to the CPU or the DTC.



Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.

In block transfer mode, even if an interrupt by an extended repeat area overflow is requested during a 1-block transfer, the remaining data in the block is transferred; transfer is terminated after a block transfer.

An interrupt by an extended repeat area overflow cannot be requested in repeat-block transfer mode.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

## 16.5 Processing on DMA Transfer Error

DMA transfer error occurs with MSAU error, the Slave TrustZone Filter error, the Master MPU error, the Slave Bus Error or the Illegal Access Error. If the access error occurs during the DMA transfer, the DMAC immediately stops the transfer of error occurred channel. At this time, the ICU setting of the corresponding channel is also cleared. If there is a request other than the channel which caused the error, it will be re-arbitration as it is.

When the transfer error occurs, DMCNT.DTE of the error causing channel is set to 0. Also, the error response is informed to the DELSRn of the corresponding channel is cleared. Write back to each register is not performed. The information of the channel that caused the error is set in DMECHR.

## 16.6 Interrupts

### 16.6.1 Transfer End Interrupt

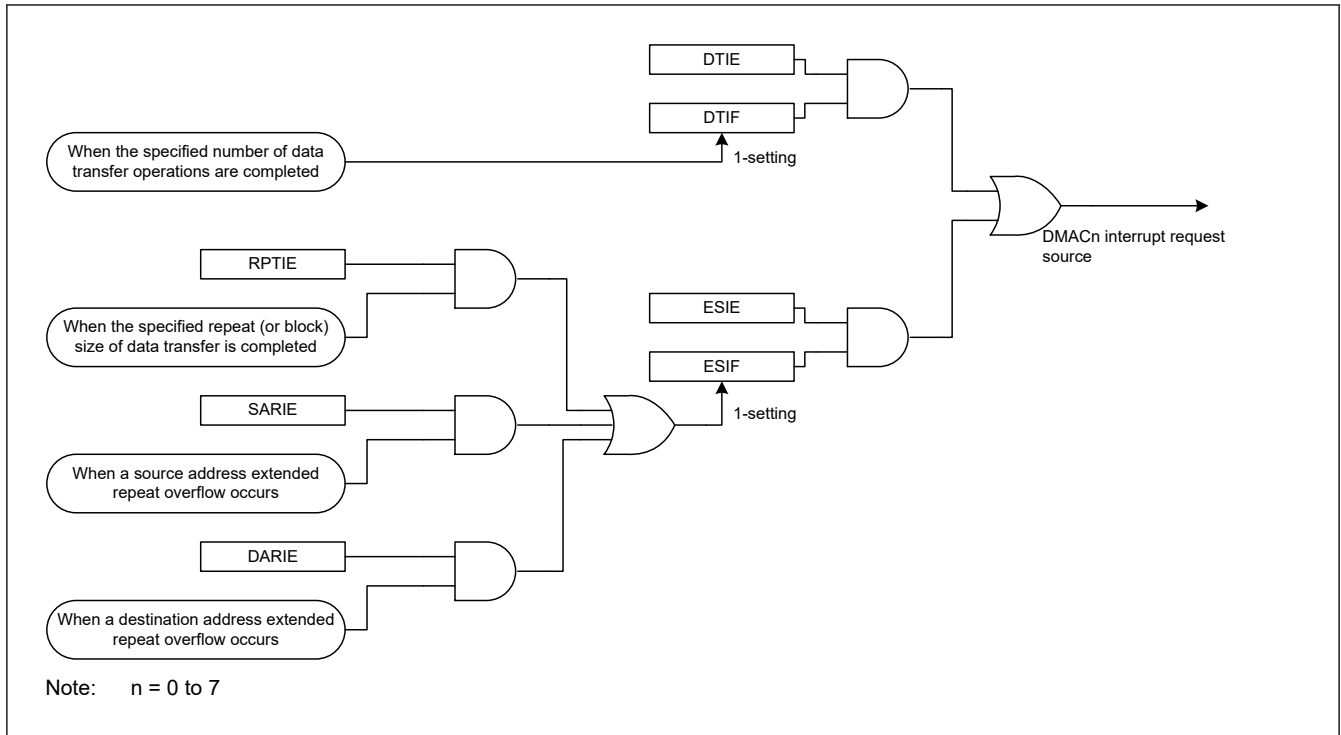
Each DMAC channel can output an interrupt request (DMAC0n\_INT) to the CPU or the DTC after transfer in response to one request is completed.

In repeat-block transfer mode, do not enable escape transfer end interrupt.

[Table 16.20](#) lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. [Figure 16.23](#) shows the schematic logic diagram of interrupt outputs (DMACn (n = 0 to 7)). [Figure 16.24](#) shows the DMAC interrupt handling routine to resume/terminate DMA transfer.

**Table 16.20 Relation among interrupt sources, interrupt status flags, and interrupt enable bits**

Interrupt sources		Interrupt enable bits	Interrupt status flags	Request output enable bits
Transfer end		—	DMSTS.DTIF	DMINT.DTIE
Escape transfer end	Repeat size end	DMINT.RPTIE	DMSTS.ESIF	DMINT.ESIE
	Source address extended repeat area overflow	DMINT.SARIE		
	Destination address extended repeat area overflow	DMINT.DARIE		



**Figure 16.23 Schematic logic diagram of interrupt output source (DMACn)**

Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases:

- When terminating a DMA transfer
- When continuing a DMA transfer

### 16.6.1.1 When Discontinuing or Terminating a DMA Transfer

Write 0 to the DMSTS.DTIF flag to clear a transfer end interrupt, and to the DMSTS.ESIF flag to clear a repeat size interrupt and an extended repeat area overflow interrupt. The corresponding DMACn channel remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DMCNT.DTE bit to 1 (DMA transfer enabled).

### 16.6.1.2 When Continuing a DMA Transfer

Write 1 to the DMCNT.DTE bit. The DMSTS.ESIF flag is automatically cleared to 0 (interrupt source cleared), and DMA transfer is resumed.

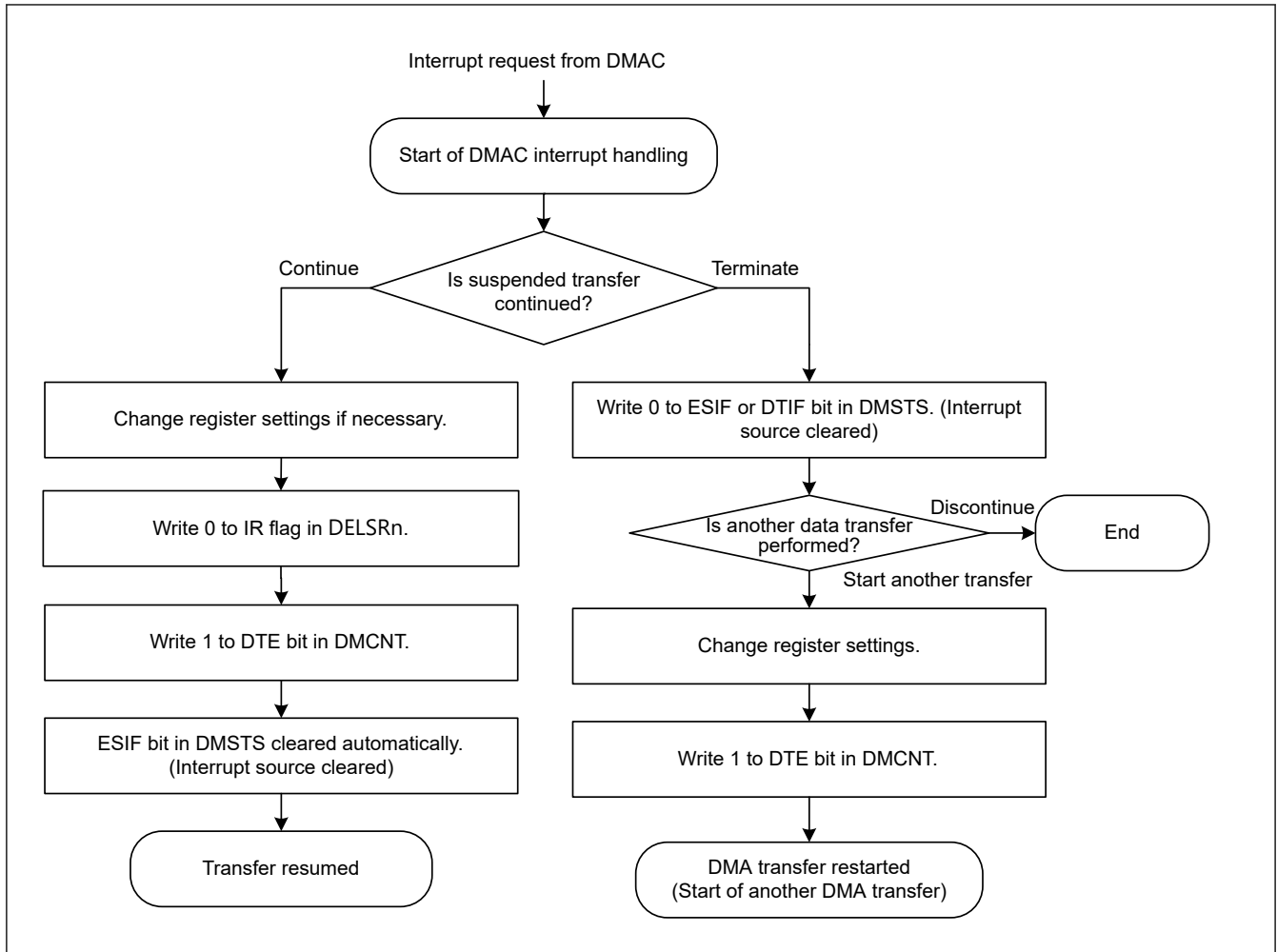


Figure 16.24 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

### 16.6.2 Transfer Error Interrupt

Error response detection interrupt request (DMA0\_TRANSERR) is generated from the DMAC/DTC when the transfer error is detected during DMAC transfer.

The types of interrupts that occur when a DMAC transfer error occurs are listed in the Table 16.21. The Table 16.21 also shows error information stored when a transfer error occurs.

Table 16.21 Interrupt and error information due to DMAC transfer error cause

Transfer error factor	NMI/RESET*1 Request	Interrupt Request	Bus Error Status	Error Address Error R/W	Error Channel Information
MSAU Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. MSERRSTAT	BUS.BMSA4ERRADD BUS.BMSA4ERRRW	DMAC. DMECHR
Master MPU Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. MMERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DMAC. DMECHR
Illegal Access Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. ILERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DMAC. DMECHR
Slave Bus Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. SLERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DMAC. DMECHR
TrustZone Filter error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. STERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DMAC. DMECHR

Note 1. If ICU.NMIER.BUSEN is enabled with DMA0\_TRANSERR set in ICU.IELSR, NMI and an interrupt will occur due to a transfer error caused by DMA. Only NMI can be generated by not setting DMA\_TRANSERR to ICU.IELSR.

## 16.7 Event Link

Each DMAC channel outputs an event link request signal (DMAC0n\_INT) every time it completes a data transfer, or a block transfer in block transfer mode.

For details, see [section 18, Event Link Controller \(ELC\)](#).

## 16.8 Low-Power Consumption Function

Before entering the module-stop state, Software Standby mode or Deep Software Standby mode, you must first set the DMAST.DMST bit to 0 (the DMAC module suspended) and use the settings in the sections that follow.

### (1) Module-stop function

Writing 1 to the MSTPCRA.MSTPA22 bit enables the module-stop function of the DMAC. If a DMA transfer is in progress when 1 is written to the MSTPA22 bit, the transition to the module-stop state proceeds after the DMA transfer ends. Access to the DMAC registers is prohibited while the MSTPA22 bit is 1. Writing 0 to the MSTPA22 bit releases the DMAC from the module-stop state.

### (2) Software Standby mode and Deep Software Standby mode

Use the settings described in [section 10.6.3.1. Transition to Software Standby Mode](#), or in [section 10.6.4.1. Transition to Deep Software Standby Mode](#).

If DMA transfer operations are in progress when the WFI instruction is executed, the DMA transfer completes before the transition to Software Standby mode or Deep Software Standby mode.

### (3) Notes on low power consumption function

For information on the WFI instruction and register settings, see [section 10.7.7. Timing of WFI Instructions](#).

To perform a DMA transfer after returning from a low power mode, set the DMAST.DMST bit to 1 again. To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DMAC startup request, specify the CPU as the interrupt request destination, as described in [section 13.5.5. Selecting Interrupt Request Destinations](#), and then execute the WFI instruction.

## 16.9 Usage Notes

### 16.9.1 Access to the Registers during DMA Transfer

Do not write to the following registers while the DMSTS.ACT flag of the same channel is set to 1 (DMAC active state) or the DMCNT.DTE bit of the same channel is set to 1 (DMA transfer enabled):

- DELSR
- DMSAR
- DMDAR
- DMCRA
- DMCRB
- DMTMD
- DMINT
- DMAMD
- DMOFR
- DMSBS
- DMDBS
- DMSRR
- DMDRR
- DMBWR
- ICUSARC

- DMAC SAR

## 16.9.2 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see [section 4, Address Space](#).

## 16.9.3 Setting of DMAC Event Link Setting Register of the Interrupt Controller Unit (DELSRn)

The DMAC event link setting register (DELSRn) should be set while the DMA transfer enable bit (DMCNT.DTE) is cleared to 0 (DMA transfer is disabled). Moreover, the DTC activation enable register (ICU.IELSRn.DTCE (n = 0 to 95)) that corresponds to the same event number that has been set by the DELSRn register should not be set to 1. For details on the ICU.IELSRn.DTCE, see [section 13, Interrupt Controller Unit \(ICU\)](#).

## 16.9.4 Suspending or Restarting DMAC Activation

To suspend a DMAC activation request, write 0x00 to the DMAC Event Link select bits (DELSRn.DELS[8:0]). To restart the DMA transfer, write the event number to the DELSRn.DELS[8:0] bits following the settings shown in [section 16.3.9, Activating the DMAC](#).

## 16.9.5 Precautions for Resuming DMA Transfer

A DMAC activation request might occur in the next request after a DMA transfer completes. If this happens, the DMA transfer starts and the DMAC activation request is held in the DMAC. To prevent this, stop the DMAC activation requests by setting the DELSRn.DELS[8:0] bits to 0.

When a DMAC activation request occurs after the last round of the DMA transfer is generated, clear the DMAC activation request with either of the following approaches.

- Clear the DMAC activation request with a DMA dummy transfer.
- Set the DMCNT.DTE bit to 0 and then set the DELSRn.IR flag to 0.

See [Figure 16.25](#).

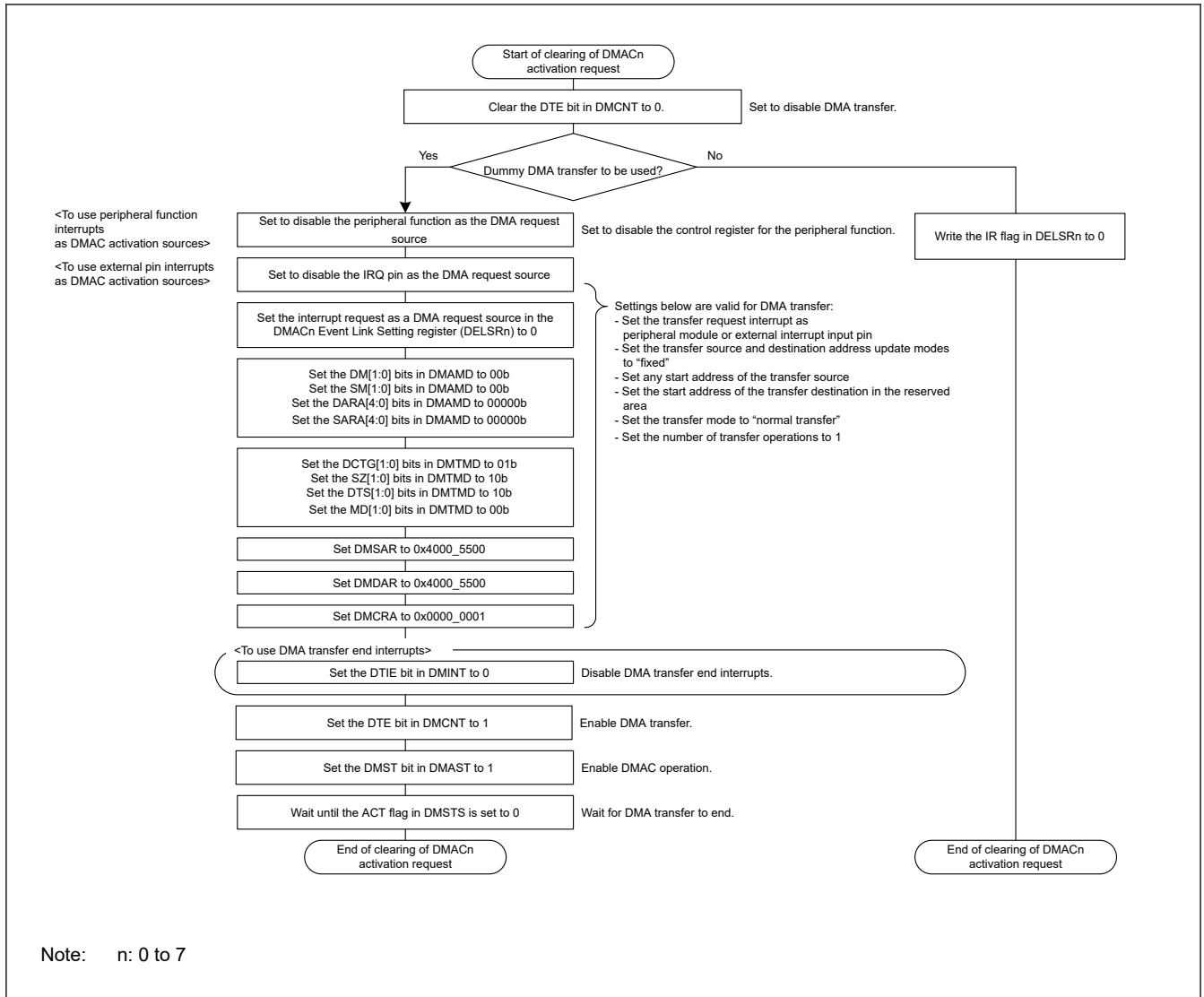


Figure 16.25 Example of register setting procedure to clear the DMAC activation interrupt

### 16.9.6 Restriction for OSPI

There is a restriction to using OSPI, see [section 35.3.8.5. Restriction in 8D-8D-8D profile 1.0 format.](#)

## 17. Data Transfer Controller (DTC)

### 17.1 Overview

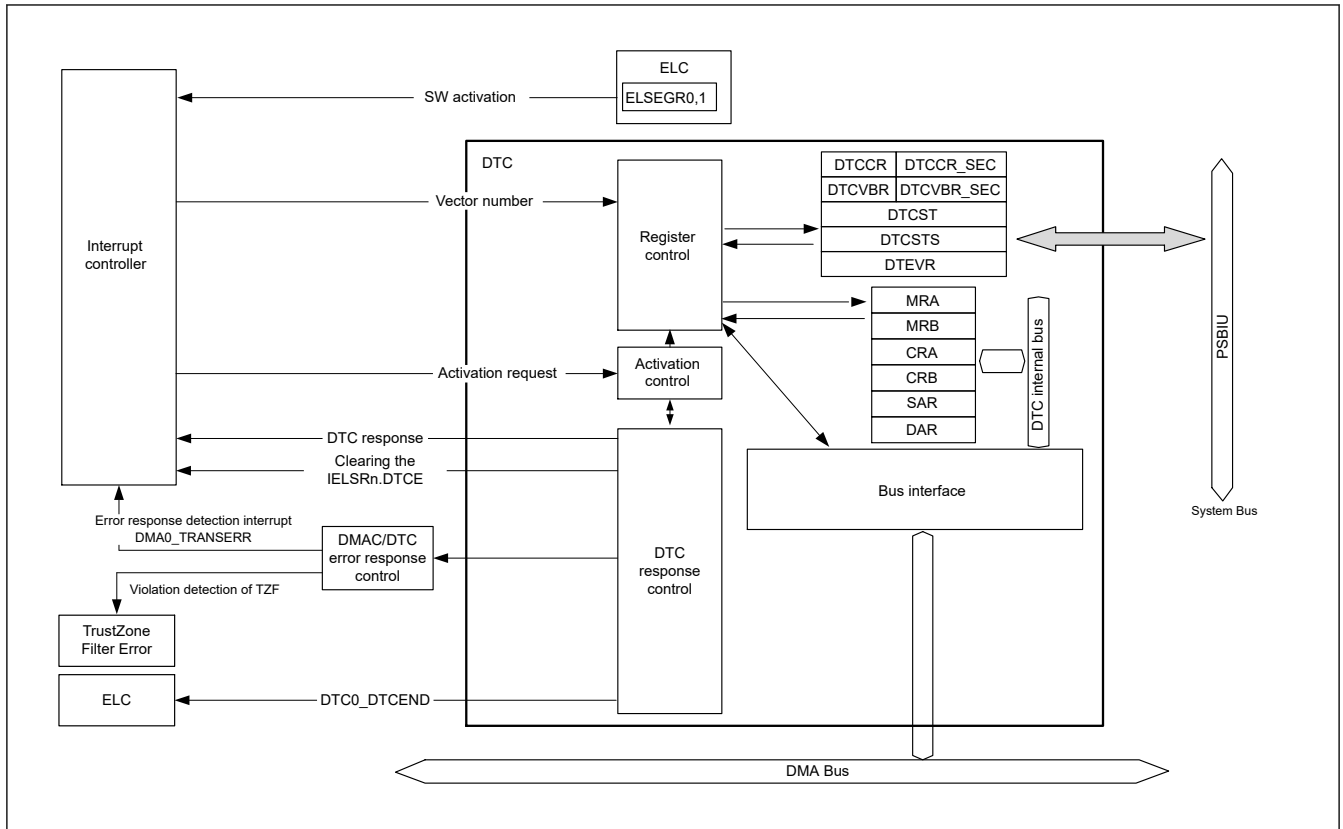
A Data Transfer Controller (DTC) module is provided for transferring data when activated by an interrupt request.

[Table 17.1](#) lists the DTC specifications and [Figure 17.1](#) shows DTC block diagram.

**Table 17.1 DTC specifications**

Parameter	Description
Transfer modes	<ul style="list-style-type: none"> <li>Normal transfer mode A single activation leads to a single data transfer.</li> <li>Repeat transfer mode A single activation leads to a single data transfer. The transfer address returns to the start address after the number of data transfers reaches the specified repeat size. The maximum number of repeat transfers is 256 and the maximum data transfer size is 256 × 32 bits (1024 bytes)</li> <li>Block transfer mode A single activation leads to a transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.</li> </ul>
Transfer channel	<ul style="list-style-type: none"> <li>Channel transfer can be associated with the interrupt source (transferred by a DTC activation request from the ICU)</li> <li>Multiple data units can be transferred on a single activation source (chain transfer)</li> <li>Chain transfers are selectable to either execute when the counter is 0, or always execute.</li> </ul>
Transfer space	<ul style="list-style-type: none"> <li>4 GB area from 0x0000_0000 to 0xFFFF_FFFF, excluding reserved areas</li> </ul>
Data transfer units	<ul style="list-style-type: none"> <li>Single data unit: 1 byte (8 bits), 1 halfword (16 bits), 1 word (32 bits)</li> <li>Single block size: 1 to 256 data units.</li> </ul>
CPU interrupt source	<ul style="list-style-type: none"> <li>An interrupt request can be generated to the CPU on a DTC activation interrupt</li> <li>An interrupt request can be generated to the CPU after a single data transfer</li> <li>An interrupt request can be generated to the CPU after a data transfer of a specified volume.</li> </ul>
Processing on DTC transfer error	<ul style="list-style-type: none"> <li>When the DTC transfer error occurs, it stops the transfer that caused the error</li> <li>Request to clear the register for activation request of DTC error number to ICU</li> </ul>
Error response detection interrupt	Generated when the DTC transfer error occurs
Event link function	An event link request is generated after one data transfer (for block, after one block transfer)
Read skip	Read of transfer information can be skipped
Write-back skip	When the transfer source or destination address is specified as fixed, a write-back of transfer information can be skipped
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set for each activation source

Note: Security attribution Register of DTC is described in ICU.ICUSARG, ICU.ICUSARH and ICU.ICUSARI



**Figure 17.1 DTC block diagram**

See [section 13.1. Overview](#) in [section 13, Interrupt Controller Unit \(ICU\)](#) for the connections between the DTC and NVIC in the CPU.

## 17.2 Register Descriptions

MRA, MRB, SAR, DAR, CRA, and CRB are all DTC internal registers that cannot be directly accessed from the CPU. Values to be set in these DTC internal registers are placed in the SRAM area as transfer information. When an activation request is generated, the DTC reads the transfer information from the SRAM area and sets it in its internal registers. After the data transfer ends, the internal register contents are written back to the SRAM area as transfer information.

### 17.2.1 DTCSAR : DTC Controller Security Attribution Register

Base address: CPSCU = 0x4000\_8000  
CPSCU\_NS = 0x5000\_8000

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTCS TSA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DTCS TSA	DTC Security Attribution 0: Secure. 1: Non-secure.	R/W



Bit	Symbol	Function	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

This register only sets the DTCST security attribute.

### DTCSTSA bit (DTC Security Attribution)

Security attributes of registers for DTCST and DTCSTS.

Do not write to the DTCSTSA bit while DTC transfer is enabled or a bus master is writing to the DTC registers.

## 17.2.2 MRA : DTC Mode Register A

Base address: DTCVBR\_SEC  
DTCVBR

Offset address:  $0x03 + 0x4 \times \text{Vector number}$   
(Inaccessible directly from the CPU. See [section 17.3.1. Allocating Transfer Information and DTC Vector Table.](#))

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MD[1:0]		SZ[1:0]		SM[1:0]		—	—
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
1:0	—	The read values are undefined. The write value should be 0.	—
3:2	SM[1:0]	Transfer Source Address Addressing Mode 0 0: Address in the SAR register is fixed (write-back to SAR is skipped.) 0 1: Address in the SAR register is fixed (write-back to SAR is skipped.) 1 0: SAR value is incremented after data transfer: +1 when SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b 1 1: SAR value is decremented after data transfer: -1 when SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b	—
5:4	SZ[1:0]	DTC Data Transfer Size 0 0: Byte (8-bit) transfer 0 1: Halfword (16-bit) transfer 1 0: Word (32-bit) transfer 1 1: Setting prohibited	—
7:6	MD[1:0]	DTC Transfer Mode Select 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

The MRA register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x03) and DTC transfers it automatically to and from the MRA register. See [section 17.3.1. Allocating Transfer Information and DTC Vector Table.](#)

### 17.2.3 MRB : DTC Mode Register B

Base address: DTCVBR\_SEC  
DTCVBR

Offset address:  $0x02 + 0x4 \times \text{Vector number}$   
(Inaccessible directly from the CPU. See [section 17.3.1. Allocating Transfer Information and DTC Vector Table.](#))

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CHNE	CHNS	DISEL	DTS	DM[1:0]	—	—	
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
1:0	—	The read values are undefined. The write value should be 0.	—
3:2	DM[1:0]	Transfer Destination Address Addressing Mode 0 0: Address in the DAR register is fixed (write-back to DAR is skipped) 0 1: Address in the DAR register is fixed (write-back to DAR is skipped) 1 0: DAR value is incremented after data transfer: +1 when MRA.SZ[1:0] = 00b +2 when SZ[1:0] = 01b +4 when SZ[1:0] = 10b 1 1: DAR value is decremented after data transfer: -1 when MRA.SZ[1:0] = 00b -2 when SZ[1:0] = 01b -4 when SZ[1:0] = 10b	—
4	DTS	DTC Transfer Mode Select 0: Select transfer destination as repeat or block area. 1: Select transfer source as repeat or block area.	—
5	DISEL	DTC Interrupt Select 0: Generate an interrupt request to the CPU when specified data transfer is complete. 1: Generate an interrupt request to the CPU each time DTC data transfer is performed.	—
6	CHNS	DTC Chain Transfer Select 0: Chain transfer is continuous. 1: Chain transfer occurs only when the transfer counter changes from 1 to 0 or 1 to CRAH.	—
7	CHNE	DTC Chain Transfer Enable 0: Chain transfer is disabled. 1: Chain transfer is enabled.	—

The MRB register cannot be accessed directly from the CPU, however the CPU can access the SRAM area (transfer information (n) start address + 0x02) and DTC transfers it automatically to and from the MRB register. See [section 17.3.1. Allocating Transfer Information and DTC Vector Table.](#)

#### DM[1:0] bits (Transfer Destination Address Addressing Mode)

The DM[1:0] bits are to fix the address of the DAR register or specify increment / decrement of the DAR register after transfer.

#### DTS bit (DTC Transfer Mode Select)

The DTS bit specifies whether the transfer source or destination is the repeat or block area in repeat or block transfer mode.

#### DISEL bit (DTC Interrupt Select)

The DISEL bit specifies the condition for generating an interrupt request to the CPU.

#### CHNS bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition. When CHNE is 0, the CHNS setting is ignored. For details on the conditions for chain transfer, see [Table 17.3.](#)

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the activation source flag is not cleared, and an interrupt request to the CPU is not generated.

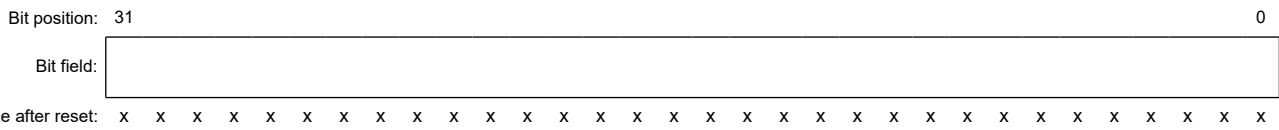
### CHNE bit (DTC Chain Transfer Enable)

The CHNE bit enables chain transfer. The chain transfer condition is selected by the CHNS bit. For details on chain transfer, see [section 17.4.6. Chain Transfer](#). When the chain transfer settings are CHNE = 1 and CHNS = 0, the transfer count according to the transfer settings will be -1 for each transfer request. Therefore, the transfer count for each transfer information must match.

### 17.2.4 SAR : DTC Transfer Source Register

Base address: DTCVBR\_SEC  
DTCVBR

Offset address: 0x04 + 0x4 × Vector number  
(Inaccessible directly from the CPU. See [section 17.3.1. Allocating Transfer Information and DTC Vector Table](#).)



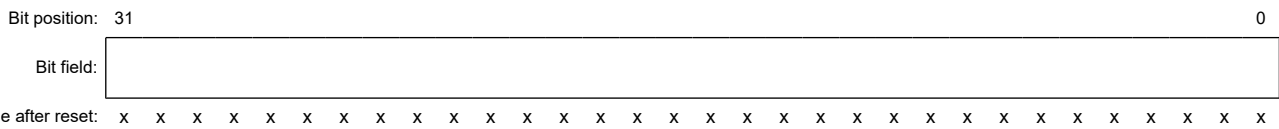
The SAR sets the transfer source start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x04) and DTC transfers it automatically to and from the SAR register. See [section 17.3.1. Allocating Transfer Information and DTC Vector Table](#).

Misalignment is prohibited for DTC transfers. Bit[0] must be 0 when MRA.SZ[1:0] = 01b, and bit[1] and bit[0] must be 0 when MRA.SZ[1:0] = 10b.

### 17.2.5 DAR : DTC Transfer Destination Register

Base address: DTCVBR\_SEC  
DTCVBR

Offset address: 0x08 + 0x4 × Vector number  
(Inaccessible directly from the CPU. See [section 17.3.1. Allocating Transfer Information and DTC Vector Table](#).)



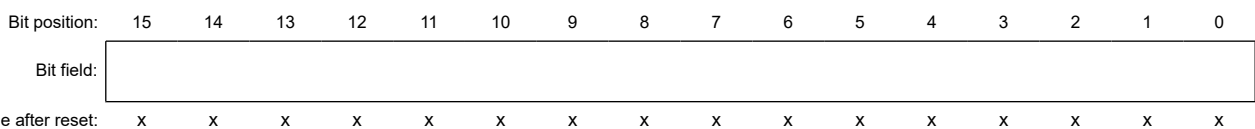
The DAR sets the transfer destination start address and cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x08) and DTC transfers it automatically to and from the DAR register. See [section 17.3.1. Allocating Transfer Information and DTC Vector Table](#).

Misalignment is prohibited for DTC transfers. Bit[0] must be 0 when MRA.SZ[1:0] = 01b, and bit[1] and bit[0] must be 0 when MRA.SZ[1:0] = 10b.

### 17.2.6 CRA : DTC Transfer Count Register A

Base address: DTCVBR\_SEC  
DTCVBR

Offset address: 0x0E + 0x4 × Vector number  
(Inaccessible directly from the CPU. See [section 17.3.1. Allocating Transfer Information and DTC Vector Table](#).)



Bit	Symbol	Function	R/W
7:0	CRAL	Transfer Counter A Lower Register Specify the transfer count.	—
15:8	CRAH	Transfer Counter A Upper Register Specify the transfer count.	—

Note: The function depends on the transfer mode.

Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

The CRA register consists of 16 bits. CRAL is the lower 8 bits and CRAH is the upper 8 bits. CRA is used in normal mode.

CRAL and CRAH are used in repeat transfer mode and block transfer mode.

The CRA register cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x0E) and DTC transfers it automatically to and from the CRA register. See [section 17.3.1. Allocating Transfer Information and DTC Vector Table](#).

### (1) Normal transfer mode (MRA.MD[1:0] = 00b)

In normal transfer mode, CRA functions as a 16-bit transfer counter. The transfer count is 1, 65535, and 65536 when the set value is 0x0001, 0xFFFF, and 0x0000, respectively. The CRA value is decremented (-1) on each data transfer.

### (2) Repeat transfer mode (MRA.MD[1:0] = 01b)

In repeat transfer mode, the CRAH register holds the transfer count and the CRAL register functions as an 8-bit transfer counter. The transfer count is 1, 255, and 256 when the set value is 0x01, 0xFF, and 0x00, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 0x00, the CRAH value is transferred to CRAL.

### (3) Block transfer mode (MRA.MD[1:0] = 10b)

In block transfer mode, the CRAH register holds the block size and the CRAL register functions as an 8-bit block size counter. The transfer count is 1, 255, and 256 when the set value is 0x01, 0xFF, and 0x00, respectively. The CRAL value is decremented (-1) on each data transfer. When it reaches 0x00, the CRAH value is transferred to CRAL.

## 17.2.7 CRB : DTC Transfer Count Register B

Base address: DTCVBR\_SEC  
DTCVBR

Offset address: 0x0C + 0x4 × Vector number  
(Inaccessible directly from the CPU. See [section 17.3.1. Allocating Transfer Information and DTC Vector Table](#).)

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Value after reset: x x x x x x x x x x x x x x x x

The CRB sets the block transfer count for block transfer mode. The transfer count is 1, 65535, and 65536 when the set value is 0x0001, 0xFFFF, and 0x0000, respectively. The CRB value is decremented (-1) when the final data of a single block size is transferred. When normal transfer mode or repeat transfer mode is selected, this register is not used, and the set value is ignored.

The CRB cannot be accessed directly from the CPU. However, the CPU can access the SRAM area (transfer information (n) start address + 0x0C) and DTC transfers it automatically to and from the CRB register. See [section 17.3.1. Allocating Transfer Information and DTC Vector Table](#).

### 17.2.8 DTCCR : DTC Control Register

Base address: DTC0\_NS = 0x5000\_AC00

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	RRS	—	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	RRS	DTC Transfer Information Read Skip Enable for Non-secure 0: Transfer information read is not skipped 1: Transfer information read is skipped when vector numbers match	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE7, P-TYPE2

#### RRS bit (DTC Transfer Information Read Skip Enable for Non-secure)

The RRS bit enables skipping of transfer information reads when vector numbers match. The DTC vector number is compared with the vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transfer information. However, when the previous transfer is a chain transfer, the transfer information is read regardless of the RRS bit.

When the transfer counter (CRA register) becomes 0 during the previous normal transfer and when the transfer counter (CRB register) becomes 0 during the previous block transfer, the transfer information is read regardless of the RRS bit value.

### 17.2.9 DTCCR\_SEC : DTC Control Register for Secure Region

Base address: DTC0 = 0x4000\_AC00

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	RRSS	—	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
4	RRSS	DTC Transfer Information Read Skip Enable for Secure 0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE6, P-TYPE2

#### RRSS bit (DTC Transfer Information Read Skip Enable for Secure)

The RRSS bit enables skipping of transfer information reads when vector numbers match. The DTC vector number is compared with the vector number in the previous activation process. When these vector numbers match and the RRSS bit is set to 1, DTC data transfer is performed without reading the transfer information. However, when the previous transfer is a chain transfer, the transfer information is read regardless of the RRSS bit.

When the transfer counter (CRA register) becomes 0 during the previous normal transfer and when the transfer counter (CRB register) becomes 0 during the previous block transfer, the transfer information is read regardless of the RRSS bit value.

### 17.2.10 DTCVBR : DTC Vector Base Register

Base address: DTC0\_NS = 0x5000\_AC00

Offset address: 0x04

Bit position: 31 0

Bit field:

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	DTC Vector Base Address for Non-secure region Set the DTC vector base address. The lower 10 bits should be 0.	R/W

Note: S-TYPE7, P-TYPE2

The DTCVBR sets the base address for calculating the DTC vector table address, which can be set in the range of 0x0000\_0000 to 0xFFFF\_FFFF (4 GB) in 1 KB units.

### 17.2.11 DTCVBR\_SEC : DTC Vector Base Register for Secure Region

Base address: DTC0 = 0x4000\_AC00

Offset address: 0x14

Bit position: 31 0

Bit field:

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	DTC Vector Base Address for Secure region Set DTC Vector Base Address for Secure region. The lower 10 bits should be 0.	R/W

Note: S-TYPE6, P-TYPE2

The DTCVBR\_SEC sets the base address for calculating the DTC vector table address, which can be set in the range of 0x0000\_0000 to 0xFFFF\_FFFF (4 GB) in 1-KB units.

### 17.2.12 DTCST : DTC Module Start Register

Base address: DTC0 = 0x4000\_AC00  
DTC0\_NS = 0x5000\_AC00

Offset address: 0x0C

Bit position: 7 6 5 4 3 2 1 0

Bit field: 

—	—	—	—	—	—	—	DTCS T
---	---	---	---	---	---	---	-----------

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	DTCST	DTC Module Start 0: DTC module stopped 1: DTC module started	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE2

**DTCST bit (DTC Module Start)**

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted. If this bit is set to 0 during a data transfer, the accepted transfer request is active until processing completes.

DTCST must be set to 0 before transitioning to one of the following state or mode:

- Module-stop state
- Software Standby mode
- Deep Software Standby mode

For details on these transitions, see [section 17.10. Low Power Consumption Function](#) and [section 10, Low Power Modes](#).

**17.2.13 DTCSTS : DTC Status Register**

Base address: DTC0 = 0x4000\_AC00  
DTC0\_NS = 0x5000\_AC00

Offset address: 0x0E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ACT	—	—	—	—	—	—	—	VECN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VECN[7:0]	DTC-Activating Vector Number Monitoring These bits indicate the vector number for the activation source when a DTC transfer is in progress. The value is only valid if a DTC transfer is in progress (ACT flag is 1).	R
14:8	—	These bits are read as 0.	R
15	ACT	DTC Active Flag 0: DTC transfer operation is not in progress. 1: DTC transfer operation is in progress.	R

Note: S-TYPE3, P-TYPE2

**VECN[7:0] bits (DTC-Activating Vector Number Monitoring)**

While transfer by the DTC is in progress, the VECN[7:0] bits indicate the vector number associated with the activation source for the transfer. The value read from the VECN[7:0] bits is valid if the ACT flag is 1, indicating a DTC transfer in progress, and invalid if the ACT flag is 0, indicating no DTC transfer is in progress.

**ACT flag (DTC Active Flag)**

The ACT flag indicates the state of the DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request.

[Clearing condition]

- When transfer by the DTC, in response to a transfer request, is complete.

### 17.2.14 DTEVR : DTC Error Vector Register

Base address: DTC0 = 0x4000\_AC00  
DTC0\_NS = 0x5000\_AC00

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DTESTA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DTEVSAM	DTEV[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DTEV[7:0]	DTC Error Vector Number These bits represent error vector of the DTC.	R
8	DTEVSAM	DTC Error Vector Number SA Monitor Indicates the SA of vector number causing the error. 0: Secure vector number 1: Non-secure vector number	R
15:9	—	These bits are read as 0. The write value should be 0.	R
16	DTESTA	DTC Error Status Flag 0: No DTC transfer error occurred 1: DTC transfer error occurred	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R

Note: P-TYPE2 See below for access by security attribution.

Note: Writing to DTESTA depends on the value of DTEVSAM

When reading this register, it can be accessed from both Secure and Non-secure access.

When writing this register, it depends on DTEVR.DTESTA.

- When DMESA = 1, it can be accessed from Secure and Non-secure access.
- When DMESA = 0, it can be accessed from Secure. An error is returned when write access is performed in a Non-secure access.

This register is cleared by a reset caused by a transfer error. Please select interrupt (DMA0\_TRANSERR) in BUS.OADCFG.OAD when you want to debug the program.

#### DTEV[7:0] bit (DTC Error Vector Number)

When a transfer error due to DTC transfer occurs, it stores the channel of DTC that was violated.

When reset was selected in MMPUOAD.OAD and MSAOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Setting condition]

- When the DTC transfer error occurs and DTESTA = 0.

[Clearing condition]

- When 1 is written to DTEVR.DTESTA.

#### DTEVSAM bit (DTC Error Vector Number SA Monitor)

When a transfer error due to DTC transfer occurs, it indicates the SA of the violating DTC vector number.

When reset was selected in MMPUOAD.OAD and MSAOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Setting condition]



- When the DTC transfer error occurs and DTESTA = 0.

[Clearing condition].

- When 1 is written to DTEVR.DTESTA.

### DTESTA bit (DTC Error Status Flag)

Indicates whether or not a DTC transfer error occurred.

DTEV, DTEVSAM, DTESTA are cleared by writing 1 to DTESTA.

Writing 0 to DTESTA is ignored.

When reset was selected in MMPUOAD.OAD and MSAOAD.OAD, Since this register is also reset. Please select NMI when you want to debug the program.

[Setting condition]

- When the DTC transfer error occurs.

[Clearing condition]

- When 1 is written to DTEVR.DTESTA.

Note: When DTEVSAM = 1, it can be cleared in the Secure state and Non-secure state. DTEVSAM = 0, it cannot be cleared in the Non-secure state.

## 17.3 Activation Sources

The DTC is activated by an interrupt request. Setting the ICU.IELSRn.DTCE bit to 1 enables activation of the DTC by the associated interrupt. The selector output  $n$  number set in ICU.IELSRn is defined as the interrupt vector number, where  $n = 0$  to 95. For an enabled interrupt, the specific DTC interrupt source associated with each interrupt vector number  $n$  is selected in ICU.IELSRn.IELS[8:0] where  $n = 0$  to 95, as listed in [section 13.3.2. Event Number](#) in [section 13, Interrupt Controller Unit \(ICU\)](#). For activation by software, see [section 18.2.2. ELSEGRn : Event Link Software Event Generation Register n \(n = 0, 1\)](#).

The interrupt vector number is equivalent to the DTC vector table number. After the DTC accepted an activation request, it does not accept another activation request until the transfer for that single request is complete, regardless of the priority of the requests. When multiple activation requests are generated during a DTC transfer, the highest priority request is accepted on completion of the transfer. When multiple activation requests are generated while the DTC Module Start bit (DTCST.DTCST) is 0, the DTC accepts the highest priority request when DTCST.DTCST is subsequently set to 1. The smaller interrupt vector number has higher priority.

The DTC performs the following operations at the start of a single data transfer or for a chain transfer, after the last of the consecutive transfers:

- On completion of a specified round of data transfer, the ICU.IELSRn.DTCE bit is set to 0, and an interrupt request is sent to the CPU.
- If the MRB.DISEL bit is 1, an interrupt request is sent to the CPU on completion of a data transfer.
- For other transfers, the ICU.IELSRn.IR flag of the activation source is set to 0 at the start of the data transfer.

### 17.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information associated with each activation source from the vector table and reads the transfer information starting at that address.

DTC has two vector tables, Non-secure side or Secure side. Because the interrupt vector number that serves as a trigger for DTC is divided into Non-secure or Secure. Place the vector table of the interrupt vector number of SA = 1 in DTCVBR which is the Non-secure side. Place the vector table of interrupt number SA = 0 in DTCVBR\_SEC which is the Secure side.

The vector table must be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC Vector Base Register (DTCVBR) to set the base address of the DTC vector table. Transfer information is allocated in the SRAM area. In the SRAM area, the start address of the transfer information  $n$  with vector number  $n$  must be  $4n$  added to the base address in the vector table.

When setting the DTC transfer attribution for the privilege access or non-privileged access, allocating the vector table and transfer information as follows. The vector table should be located in the privileged area. Transfer information with the privileged attribution is located in the privileged area. Transfer information with the unprivileged attribution is located in the Unprivileged area. Privileged area is set by MMPU (see [section 15, Memory Protection Unit \(MPU\)](#)). Privileged and non-privileged access settings are located at bit 0 of the address corresponding to each vector number in the vector table.

Figure 17.2 shows the relationship between the DTC vector table and transfer information.

Figure 17.3 shows the bit arrangement in the vector table.

Figure 17.4 shows the allocation of transfer information in the SRAM area.

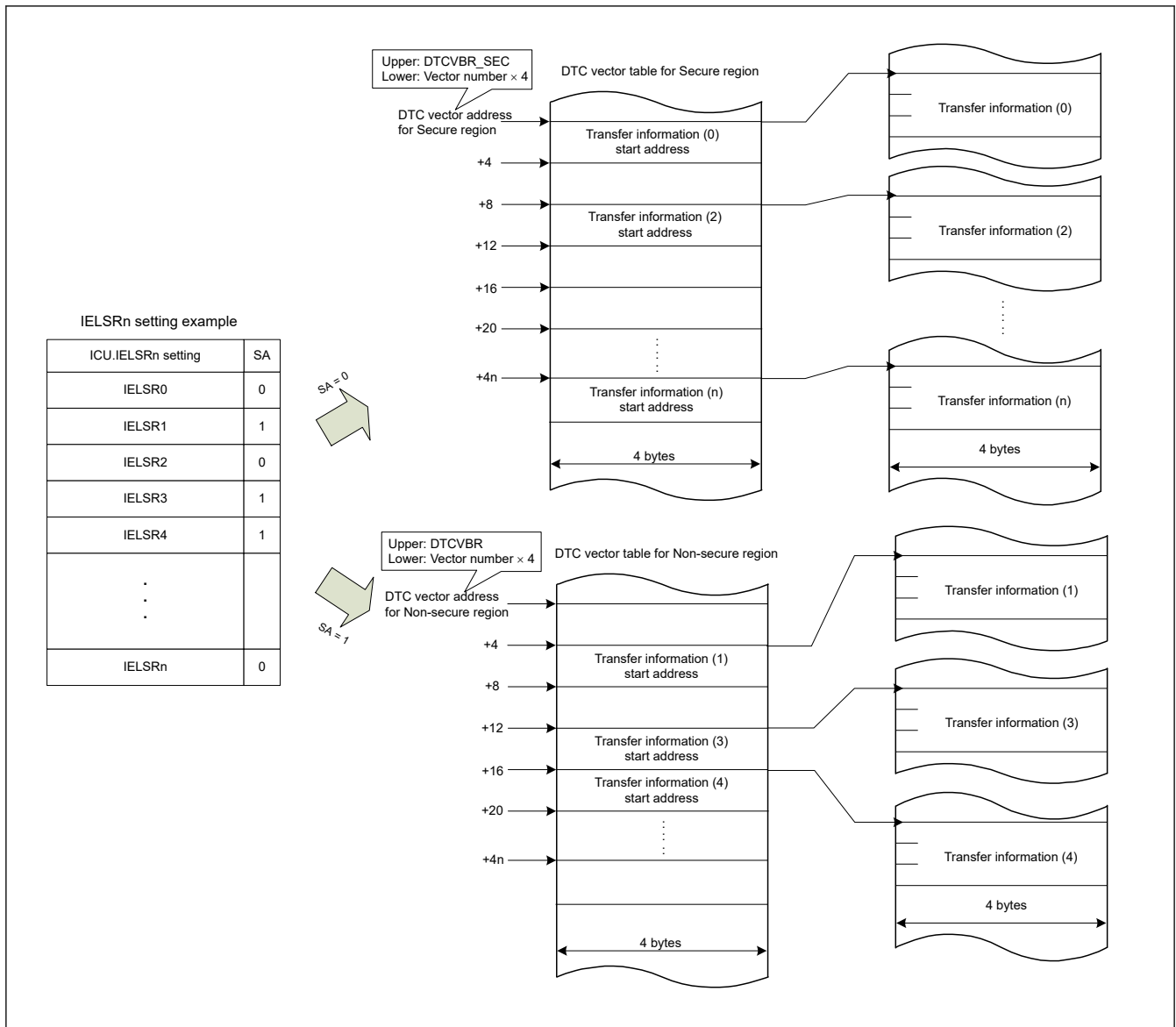


Figure 17.2 DTC vector table and transfer information

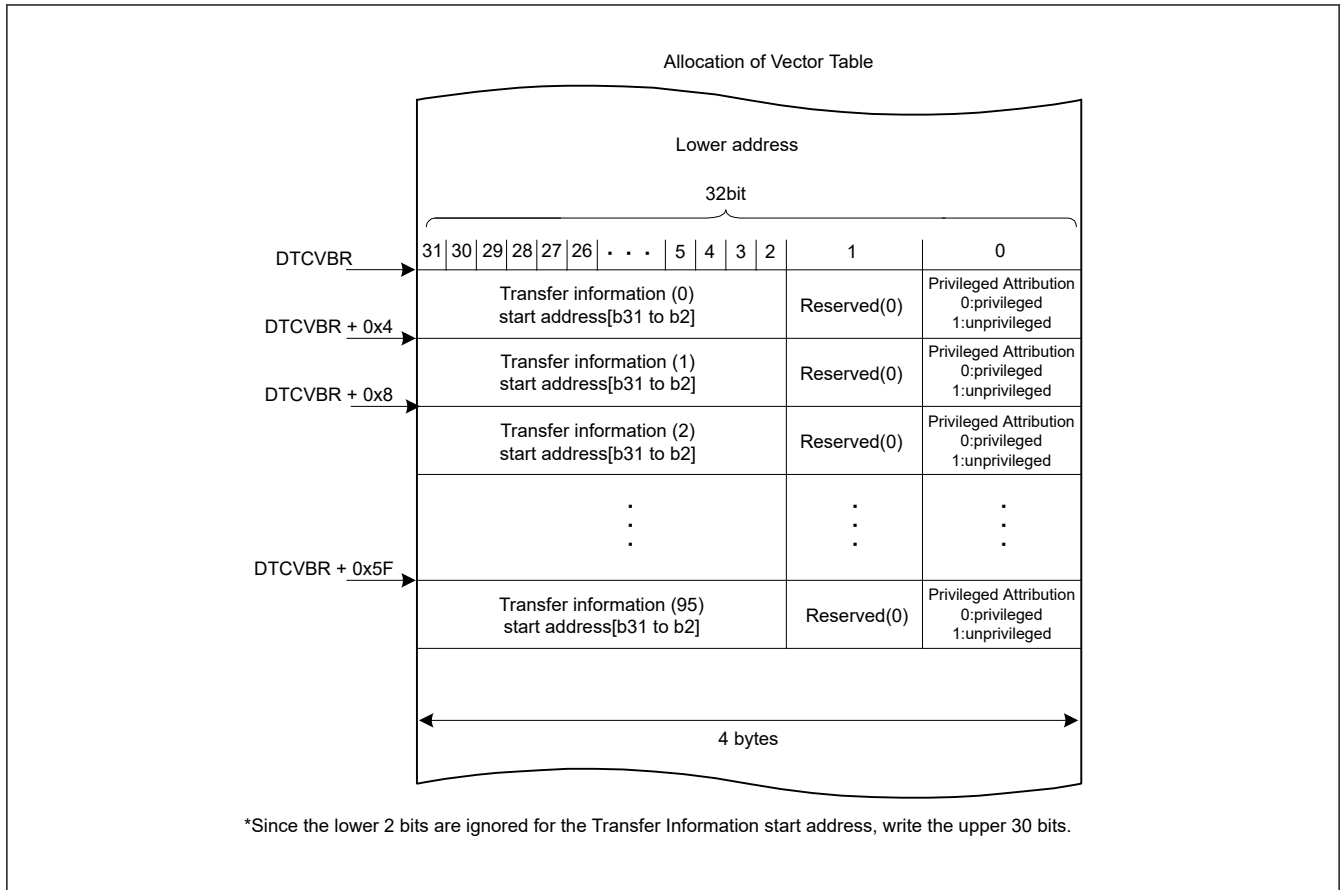


Figure 17.3 Bit arrangement in the vector table

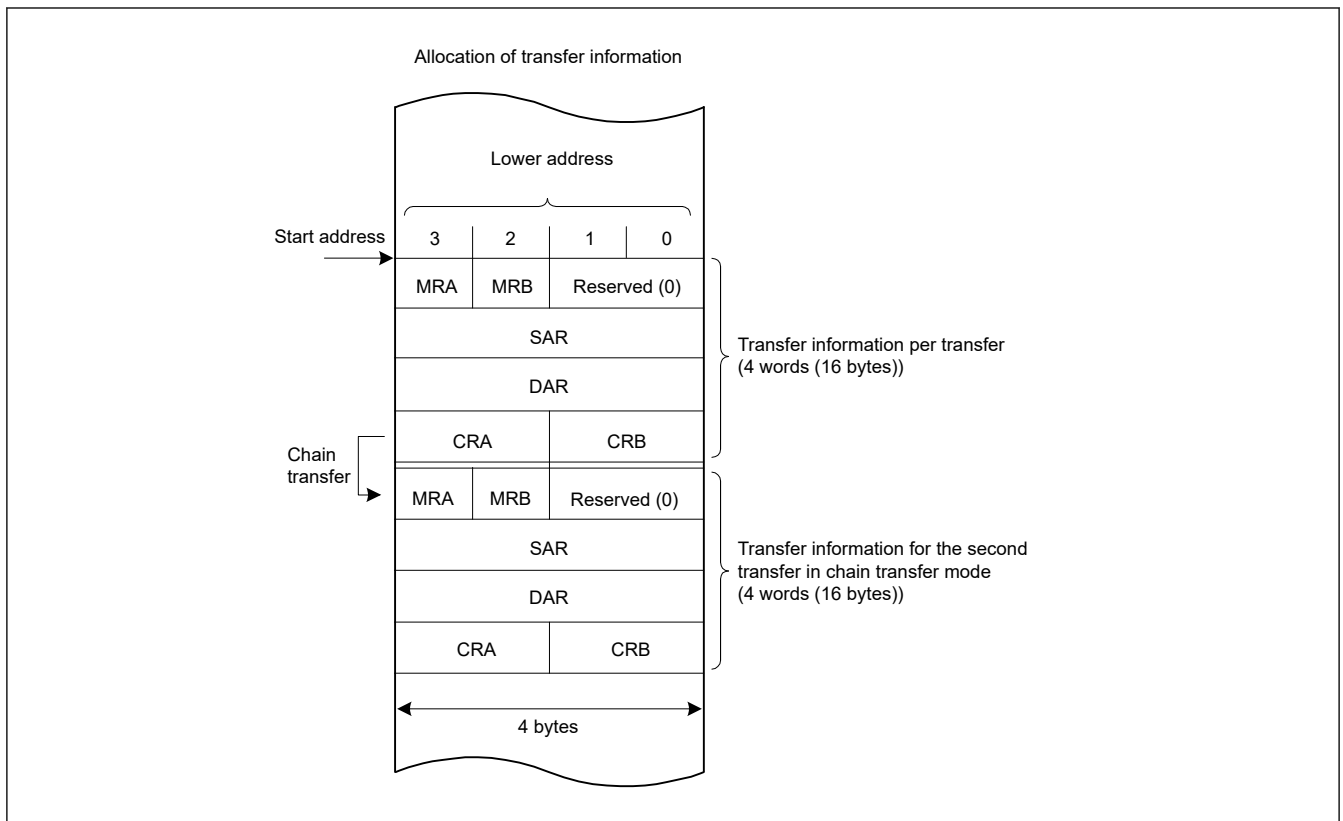


Figure 17.4 Allocation of transfer information in the SRAM area

## 17.4 Operation

The DTC transfers data according to the transfer information. Storage of the transfer information in the SRAM area is required before a DTC operation. When the DTC is activated, it reads the DTC vector associated with the vector number. The DTC reads the transfer information from the transfer information store address referenced by the DTC vector and transfers the data. After the data transfer, the DTC writes back the transfer information. Storing the transfer information in the SRAM area allows data transfer of any number of channels.

The transfer modes include:

- Normal transfer mode
- Repeat transfer mode
- Block transfer mode.

The DTC specifies a transfer source address in the SAR register and a transfer destination address in the DAR register. The values of these registers are incremented, decremented, or address-fixed independently after the data transfer.

Table 17.2 describes the DTC transfer modes.

**Table 17.2 DTC transfer modes**

Transfer mode	Data size transferred on single transfer request	Increment or decrement of memory address	Settable transfer count
Normal transfer mode	1 byte (8 bit), 1 halfword (16 bit), 1 word (32 bit)	Incremented or decremented by 1, 2, or 4 or address-fixed	1 to 65536
Repeat transfer mode*1	1 byte (8 bit), 1 halfword (16 bit), 1 word (32 bit)	Incremented or decremented by 1, 2, or 4 or address-fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes))	Incremented or decremented by 1, 2, or 4 or address-fixed	1 to 65536

Note 1. Set the transfer source or transfer destination as the repeat area.

Note 2. Set the transfer source or transfer destination as the block area.

Note 3. After a data transfer of the specified count, the initial state is restored and operation restarts.

Setting the MRB.CHNE bit to 1 allows multiple transfers or chain transfer on a single activation source. It also enables a chain transfer when the specified data transfer is complete.

Figure 17.5 shows the operation flow of the DTC. Table 17.3 lists the chain transfer conditions. The combination of control information for the second and subsequent transfers are omitted in this table.

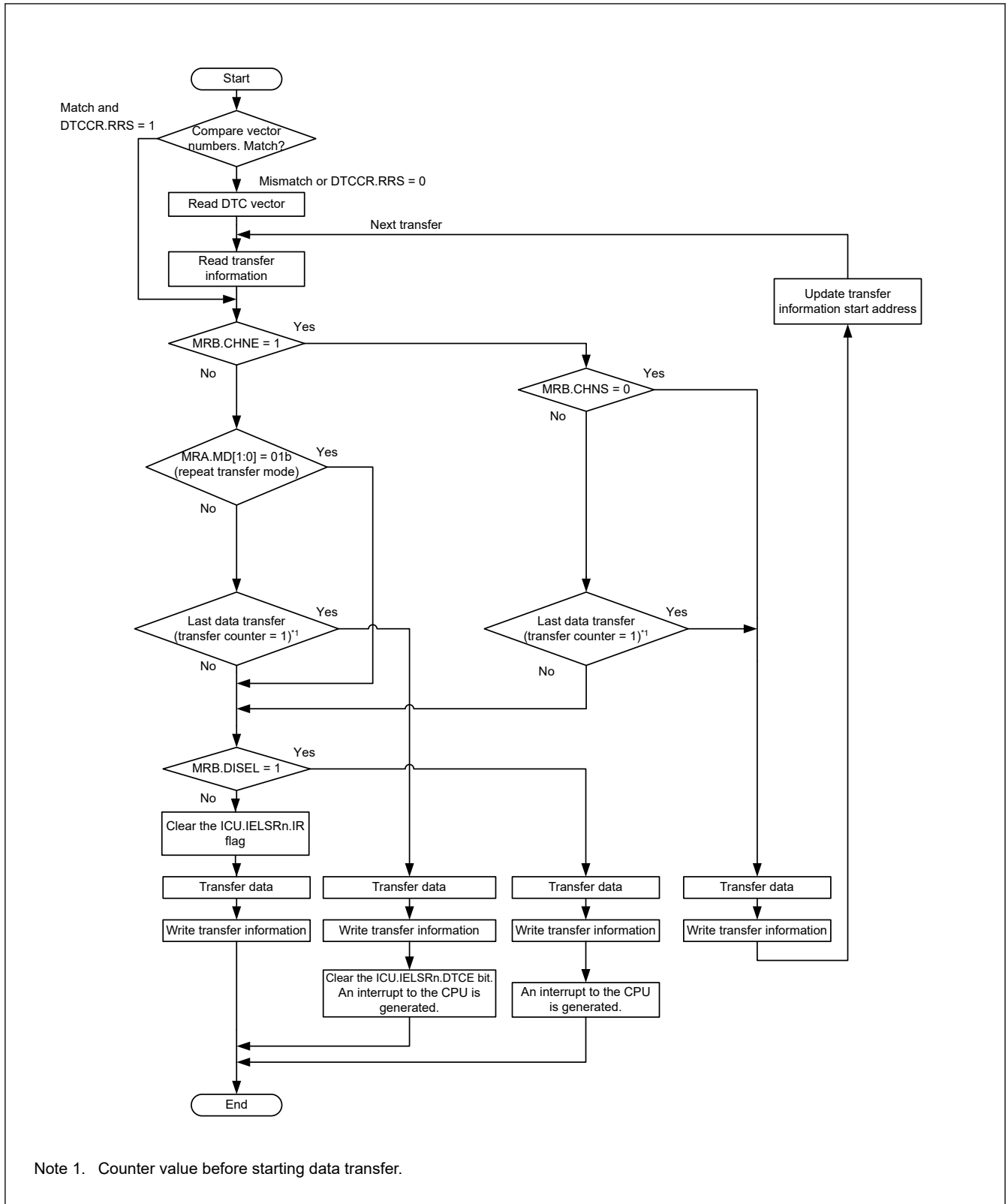


Figure 17.5 DTC operation flow

**Table 17.3 Chain transfer conditions**

First transfer				Second transfer <sup>*3</sup>				DTC transfer
CHNE bit	CHNS bit	DISEL bit	Transfer counter <sup>*1 *2</sup>	CHNE bit	CHNS bit	DISEL bit	Transfer counter <sup>*1 *2</sup>	
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → *)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → *)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → *)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counter used depends on the transfer modes as follows:

- Normal transfer mode — CRA register
- Repeat transfer mode — CRAL register
- Block transfer mode — CRB register

Note 2. On completion of a data transfer, the counters operate as follows:

- 1 → 0 in normal and block transfer modes
- 1 → CRAH in repeat transfer mode
- (1 → \*) in the table indicates both of these two operations, depending on the mode.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The conditions for the combination of the second transfer and CHNE = 1 is omitted.

### 17.4.1 Transfer Information Read Skip Function

Reading of vector addresses and transfer information can be skipped by setting the DTCCR.RRS bit. When a DTC activation request is generated, the current DTC vector number is compared with the DTC vector number in the previous activation process. When these vector numbers match and the RRS bit is set to 1, the DTC data transfer is performed without reading the vector address and transfer information. However, when the previous transfer is a chain transfer, the vector address and transfer information are read. Additionally, when the transfer counter (CRA register) becomes 0 during the previous normal transfer, and when the transfer counter (CRB register) becomes 0 during the previous block transfer, transfer information is read regardless of the RRS bit. [Figure 17.13](#) shows an example when reading the transfer information is skipped.

To update the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, then set the RRS bit to 1. The stored vector number is discarded by setting the RRS bit to 0. The updated DTC vector table and transfer information are read in the next activation process.

### 17.4.2 Transfer Information Write-Back Skip Function

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to address fixed, a part of the transfer information is not written back. [Table 17.4](#) lists the transfer information write-back skip conditions and the associated registers. The CRA and CRB registers are written back, and the write-back of the MRA and MRB registers is skipped.

**Table 17.4 Transfer information write-back skip conditions and applicable registers**

MRA.SM[1:0] bits		MRB.DM[1:0] bits		SAR register	DAR register
b3	b2	b3	b2		
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

### 17.4.3 Normal Transfer Mode

The normal transfer mode allows a 1-byte (8 bit), 1-halfword (16 bit), 1-word (32 bit) data transfer on a single activation source. The transfer count can be set from 1 to 65536. Transfer source and destination addresses can be independently set to increment, decrement, or fixed. This mode enables an interrupt request to the CPU to be generated at the end of a specified-count transfer.

[Table 17.5](#) lists register functions in normal transfer mode, and [Figure 17.6](#) shows the memory map of normal transfer mode.

**Table 17.5 Register functions in normal transfer mode**

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	Increment, decrement, or fixed*1
DAR	Transfer destination address	Increment, decrement, fixed*1
CRA	Transfer counter A	CRA - 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped in address-fixed mode.

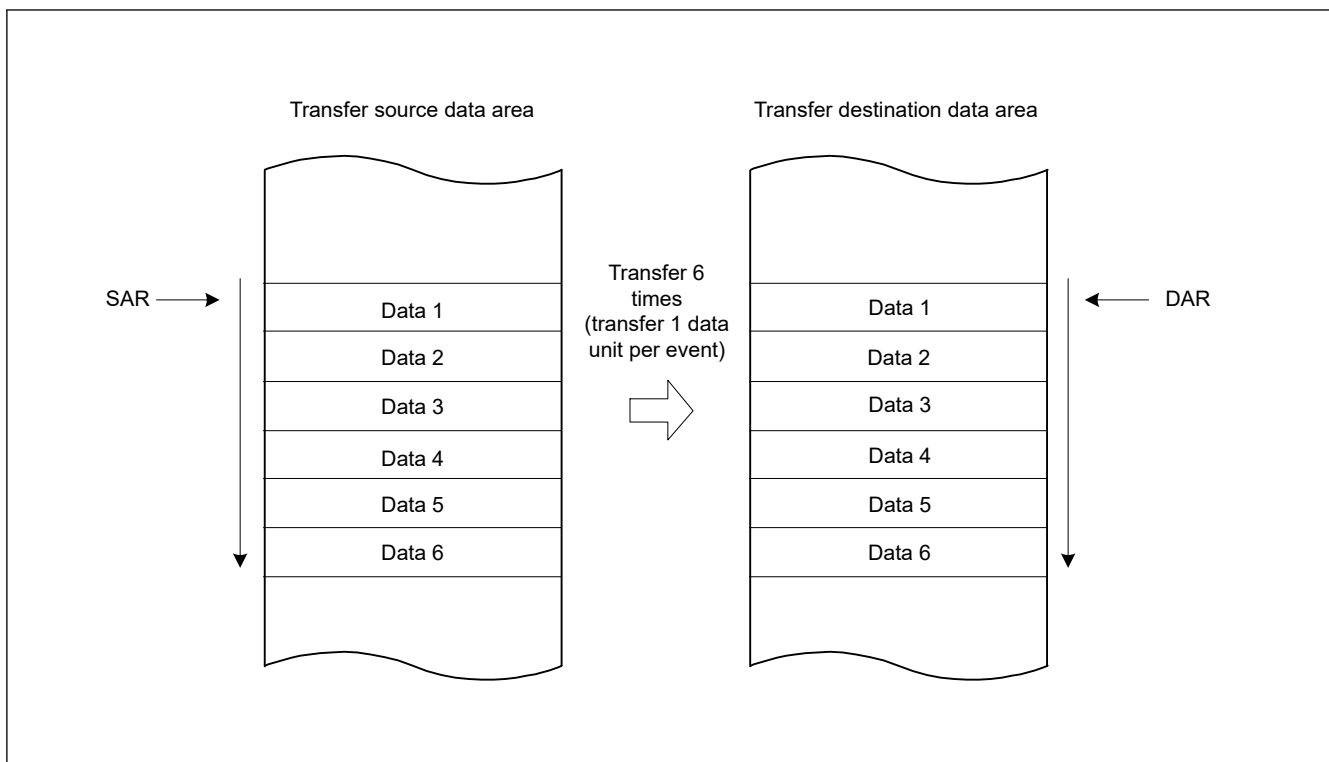


Figure 17.6 Memory map of normal transfer mode (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRA = 0x0006)

### 17.4.4 Repeat Transfer Mode

The repeat transfer mode allows a 1-byte (8-bit), 1-halfword (16-bit), or 1-word (32-bit) data transfer on a single activation source. Transfer source or transfer destination for the repeat area must be specified in the MRB.DTS bit. The transfer count can be set from 1 to 256. When the specified transfer count is complete, the initial value of the address register specified in the repeat area is restored, the initial value of the transfer counter is restored, and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.

When the transfer counter CRAL decrements to 0x00 in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. As a result, the transfer counter does not clear to 0x00, which disables interrupt requests to the CPU when the MRB.DISEL bit is set to 0. An interrupt request to the CPU is generated when the specified data transfer completes.

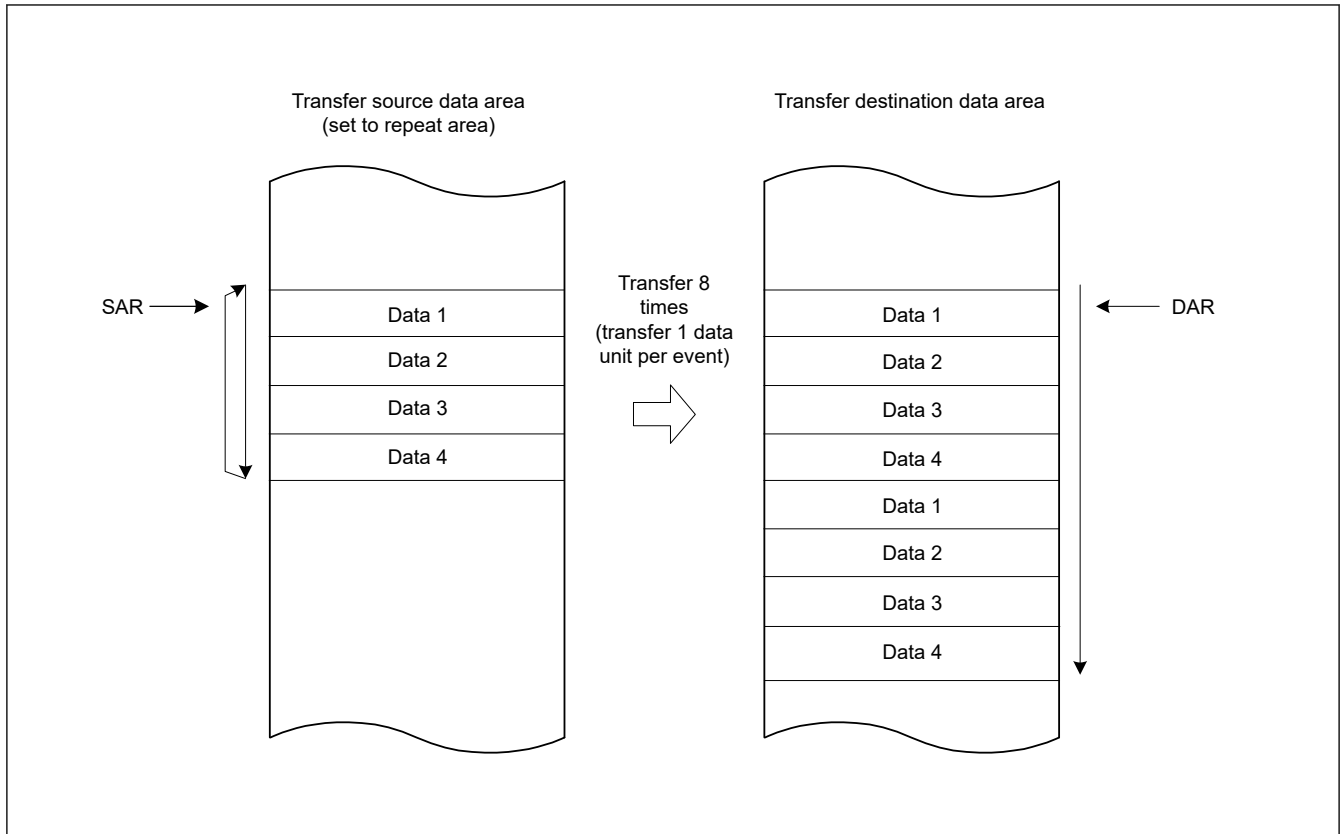
Table 17.6 lists the register functions in repeat transfer mode, and Figure 17.7 shows the memory map of repeat transfer mode.

Table 17.6 Register functions in repeat transfer mode

Register	Description	Value written back by writing transfer information	
		When CRAL is not 1	When CRAL is 1
SAR	Transfer source address	Increment, decrement, fixed <sup>*1</sup>	<ul style="list-style-type: none"> <li>When the MRB.DTS bit is 0 Increment, decrement, or fixed<sup>*1</sup></li> <li>When the MRB.DTS bit is 1 SAR register initial value</li> </ul>
DAR	Transfer destination address	Increment, decrement, or fixed <sup>*1</sup>	<ul style="list-style-type: none"> <li>When the MRB.DTS bit is 0 DAR register initial value</li> <li>When the MRB.DTS bit is 1 Increment, decrement, or fixed<sup>*1</sup></li> </ul>
CRAH	Retains transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note 1. Write-back is skipped in address-fixed mode.





**Figure 17.7** Memory map of repeat transfer mode when transfer source is a repeat area (MRA.SM[1:0] = 10b, MRB.DM[1:0] = 10b, CRAH = 0x04)

### 17.4.5 Block Transfer Mode

The block transfer mode allows single-block data transfer on a single activation source. Transfer source or transfer destination for the block area must be specified in the MRB.DTS bit. The block size can be set from 1 to 256 bytes, 1 to 256 halfwords (2 to 512 bytes), or 1 to 256 words (4 to 1024 bytes). When transfer of the specified block completes, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS = 1 or the DAR register when the DTS = 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.

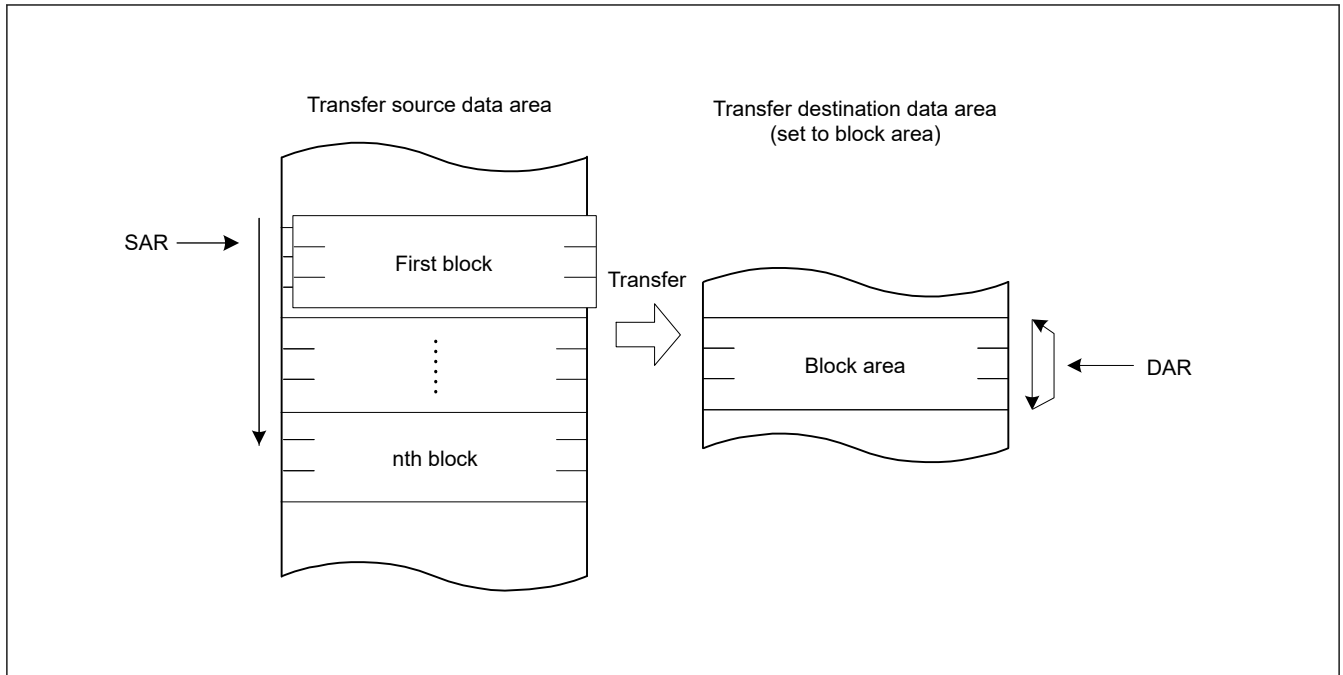
The transfer count (block count) can be set from 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of the specified-count block transfer.

Table 17.7 lists the register functions in block transfer mode, and Figure 17.8 shows the memory map for block transfer mode.

**Table 17.7** Register functions in block transfer mode

Register	Description	Value written back by writing transfer information
SAR	Transfer source address	<ul style="list-style-type: none"> <li>When MRB.DTS bit is 0 Increment, decrement, or fixed*1</li> <li>When MRB.DTS bit is 1 SAR register initial value.</li> </ul>
DAR	Transfer destination address	<ul style="list-style-type: none"> <li>When MRB.DTS bit is 0 DAR register initial value</li> <li>When MRB.DTS bit is 1 Increment, decrement, or fixed*1.</li> </ul>
CRAH	Holds block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note 1. Write-back is skipped in address-fixed mode.

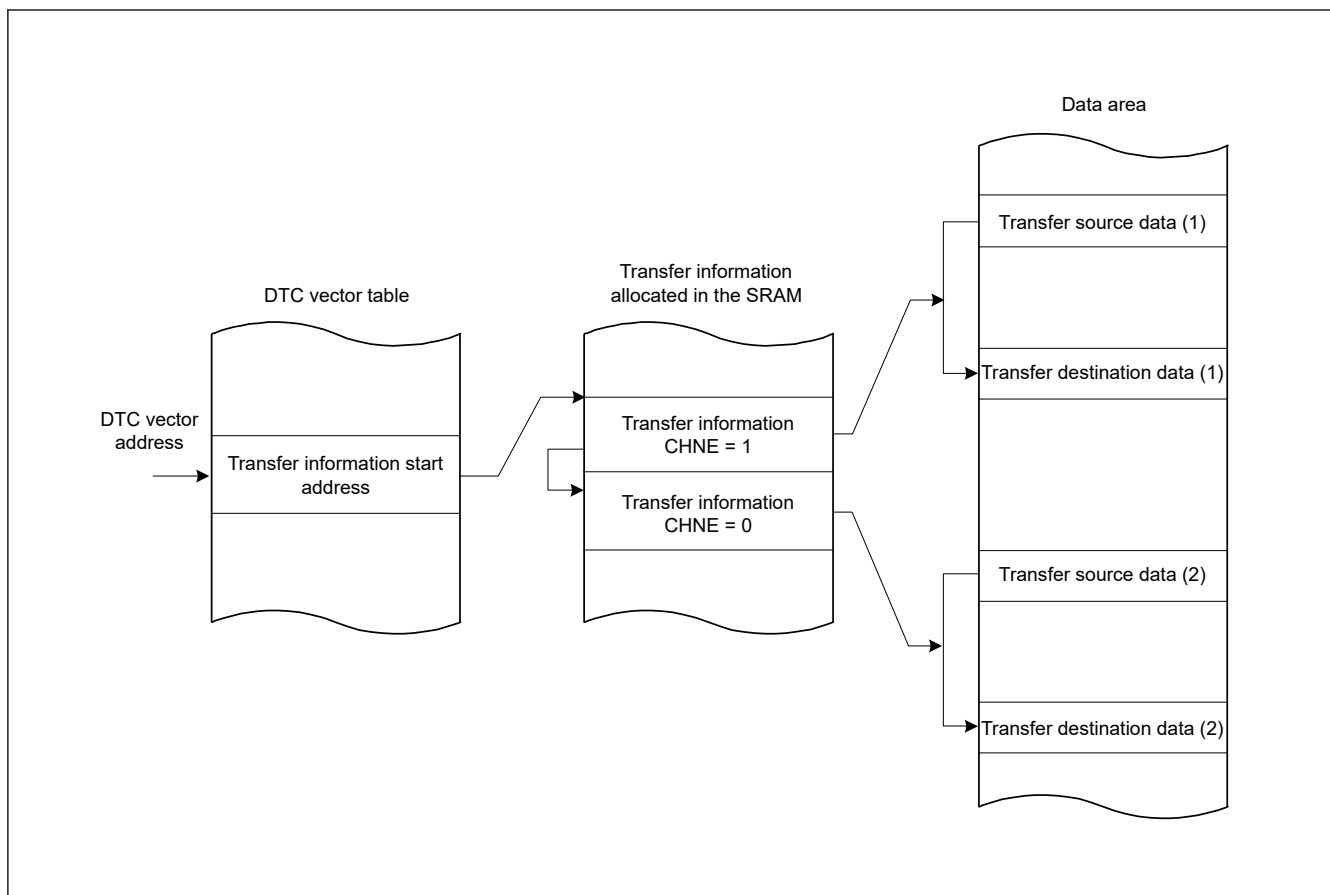


**Figure 17.8** Memory map of block transfer mode

### 17.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single activation source. If the MRB.CHNE is set to 1 and CHNS to 0, an interrupt request to the CPU is not generated on completion of the specified number of rounds of transfer or by setting the MRB.DISEL bit to 1. An interrupt request is sent to the CPU each time DTC data transfer is performed. Data transfer has no effect on the ICU.IELSRn.IR flag of the activation source.

The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define the data transfer. [Figure 17.9](#) shows a chain transfer operation.



**Figure 17.9 Chain transfer operation**

Writing 1 to the MRB.CHNE and CHNS bits enables chain transfer to be performed only after completion of the specified data transfer. In repeat transfer mode, chain transfer is performed after completion of the specified data transfer. For details on chain transfer conditions, see [Table 17.3](#).

### 17.4.7 Operation Timing

[Figure 17.10](#) to [Figure 17.13](#) are timing diagrams that show the minimum number of execution cycles.

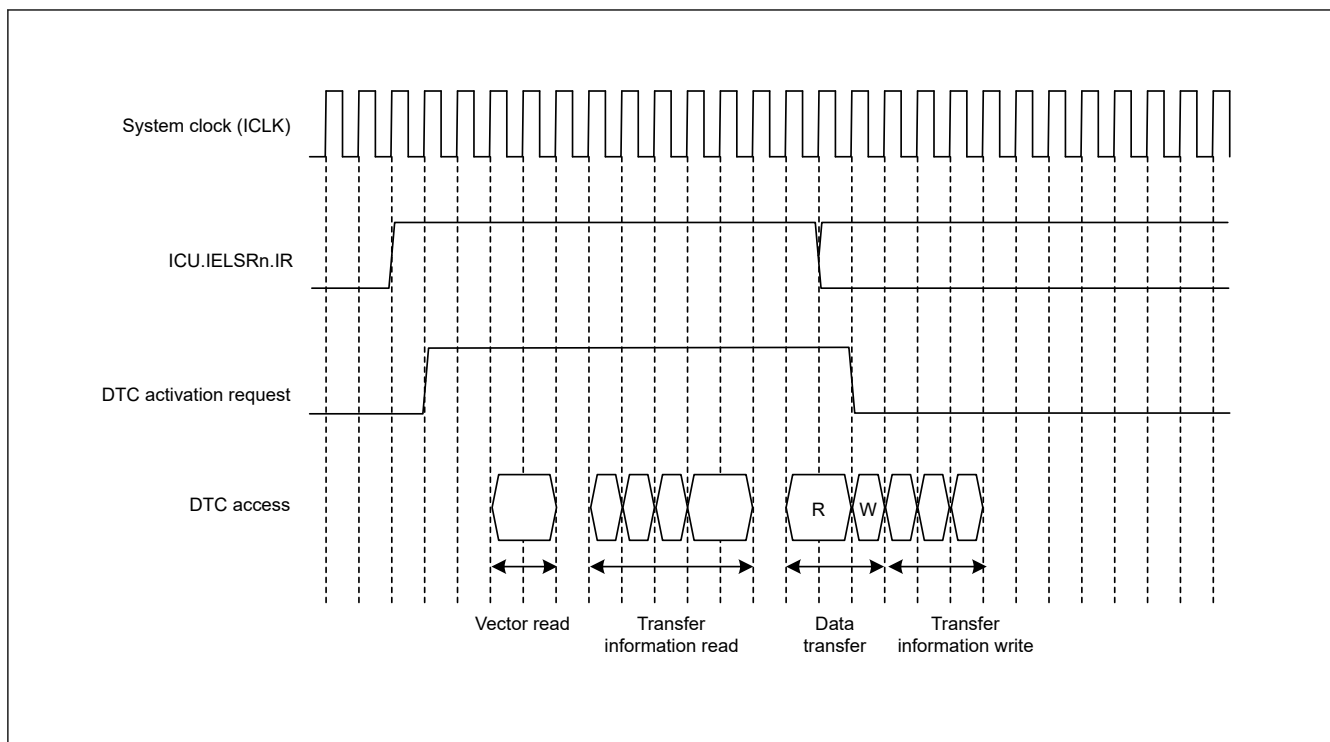


Figure 17.10 Example 1 of DTC operation timing in normal transfer and repeat transfer modes

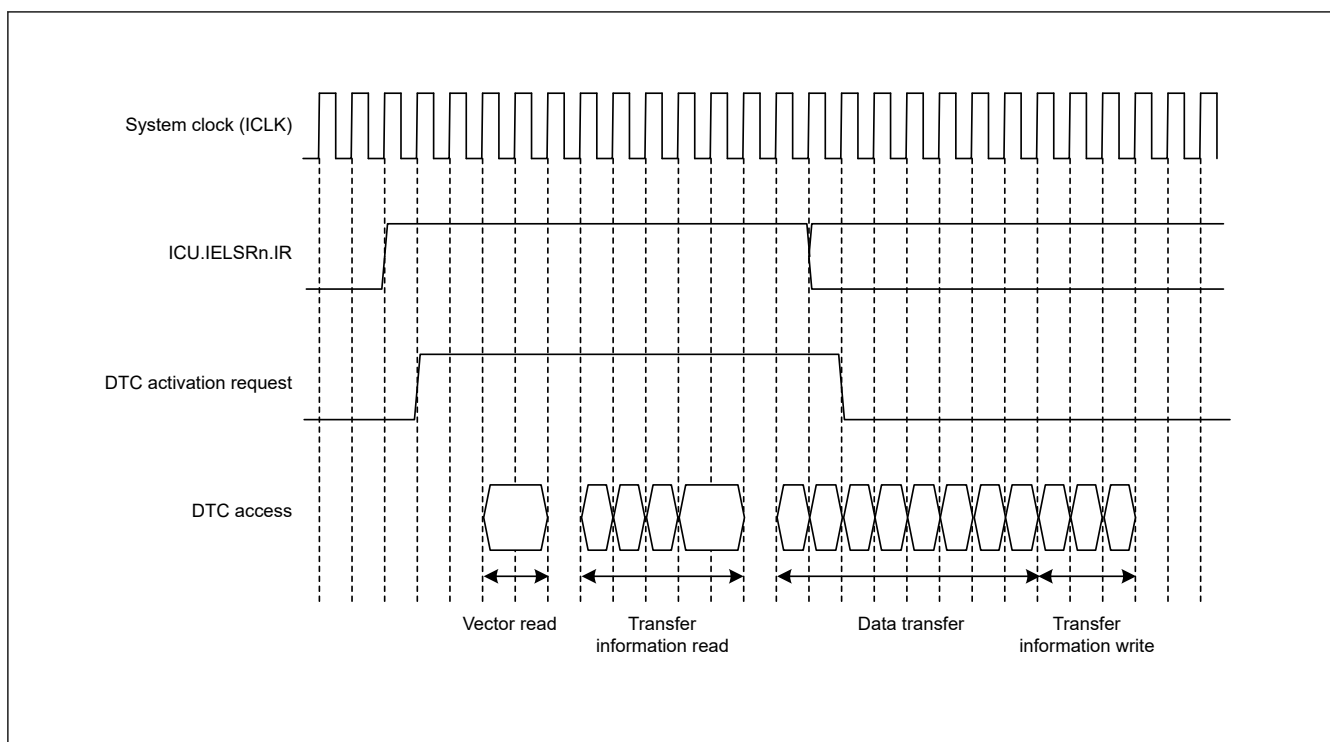


Figure 17.11 Example 2 of DTC operation timing in block transfer mode when the block size = 4

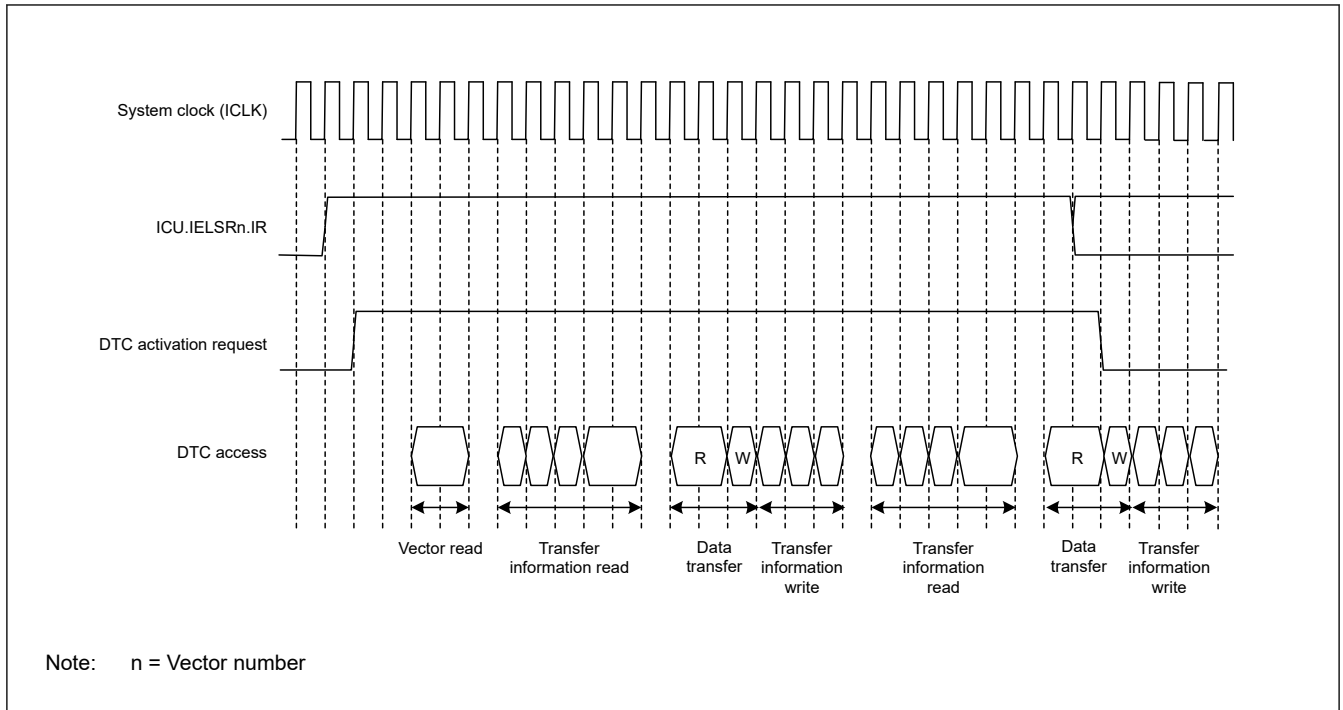


Figure 17.12 Example 3 of DTC operation timing for chain transfer

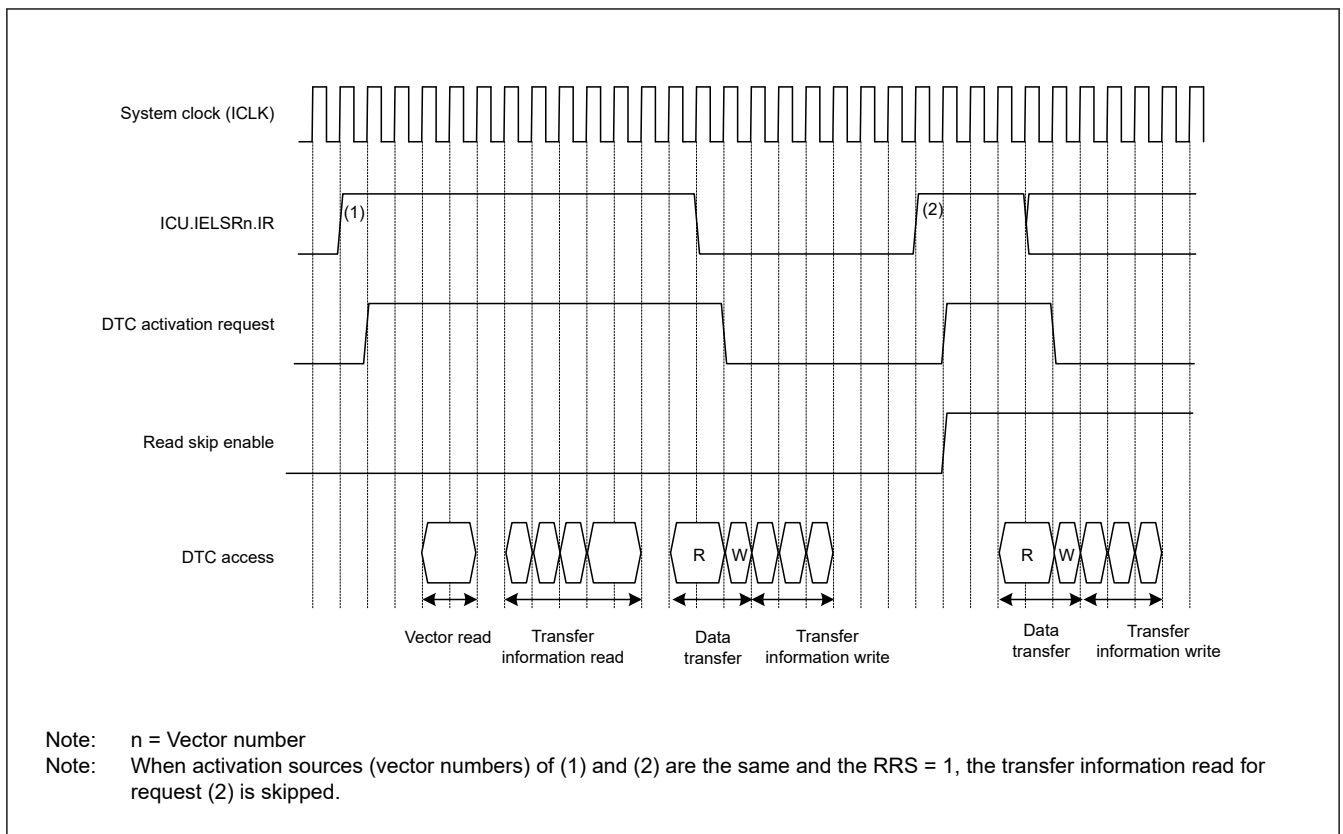


Figure 17.13 Example of operation when a transfer information read is skipped with the vector, transfer information, and transfer destination data on the SRAM, and the transfer source data on the peripheral module

### 17.4.8 Vector Security

The security attribute of transfer access of DTC vector  $n$  and security attribute of access to the IELSR $n$  ( $n = 0$  to 95) register of ICU are controlled by SAIELSR $n$  bit of ICUSAR $x$  ( $x = G, H$  or  $I$ ) registers in CPSCU. For details on the CPSCU.ICUSAR $x$  registers, see [section 13, Interrupt Controller Unit \(ICU\)](#).

When the CPSCU.ICUSAR $x$ .IELSR $n$ SA bit is 0

- IELSR $n$  register have security attribute
- The vector number of DTC activated by the interrupt request set in IELS behaves as a Secure master.

When the CPSCU.ICUSAR $x$ .IELSR $n$ SA bit is 1

- IELSR $n$  register have non-security attribute
- The vector number of DTC activated by the interrupt request set in IELS behaves as a Non-secure master.

For areas accessible to Secure or Non-secure masters, See [section 14, Buses](#), and [section 40, Security Features](#).

### 17.4.9 Vector privilege

The privileged attribute of transfer access of DTC vector  $n$  is controlled bit0 in vector table.

If bit0 corresponding to each vector number in the vector table is set to 0, DTC transfer is privileged access.

If bit0 is set to 1, DTC transfers are unprivileged access.

Refer to for vector table and transfer information.

## 17.5 DTC Setting Procedure

Before using the DTC, set the DTC Vector Base Register (DTCVBR). Set the ICU.IELSR $n$ .IELS[8:0] bits to 0 to disable the interrupt in the NVIC and follow the procedure in [Table 17.8](#) to set the DTC.

**Table 17.8 DTC setting procedure**

No.	Step Name	Description
1	Set the DTCCR <sup>*1</sup> .RRS bit to 0	Set the DTCCR <sup>*1</sup> .RRS bit to 0 to reset the transfer information read skip flag. After that, the transfer information read is not skipped while the DTC is activated. Be sure to specify this setting when the transfer information is updated.
2	Set transfer information (MRA, MRB, SAR, DAR, CRA, and CRB)	Allocate transfer information (MRA, MRB, SAR, DAR, CRA, and CRB) in the data area. To set transfer information, see <a href="#">section 17.2, Register Descriptions</a> . To allocate transfer information, see <a href="#">section 17.3.1, Allocating Transfer Information and DTC Vector Table</a> .
3	Set transfer information start addresses in the DTC vector table	Set the transfer information start addresses in the DTC vector table. To set the DTC vector table, see <a href="#">section 17.3.1, Allocating Transfer Information and DTC Vector Table</a> .
4	Set the DTCCR <sup>*1</sup> .RRS bit to 1	Set the DTCCR <sup>*1</sup> .RRS bit to 1 to enable skipping of the second and subsequent transfer information read cycles for continuous DTC activation from the same interrupt source. The RRS bit can be set to 1, but if this is set during DTC transfer, it becomes valid from the next transfer.
5	Set the ICU.IELSR $n$ .DTCE bit to 1. Set the ICU.IELSR $n$ .IELS[8:0] as interrupt source. The interrupt should be enabled in the NVIC.	Set the ICU.IELSR $n$ .DTCE bit to 1. Set ICU.IELSR $n$ .IELS[8:0] as interrupt sources that trigger DTC. The interrupt must be enabled in the NVIC. See <a href="#">section 13.3.2, Event Number</a> in <a href="#">section 13, Interrupt Controller Unit (ICU)</a> .
6	Set the enable bit for an activation source interrupt	Set the enable bit for the activation source interrupts to 1. When a source interrupt is generated, the DTC is activated. To set the interrupt source enable bit, see the settings for the modules that are to be the activation sources.
7	Set the DTCST.DTCST bit to 1	Set the DTC Module Start bit (DTCST.DTCST) to 1.

Note: The DTCST.DTCST bit can be set even if the setting for each activation source is not completed.

Note: When DTCSAR.DTCSTSA = 0 is set, DTCST.DTCST = 1 must be set in the Secure state to use DTC in the Non-secure state.

Note 1. When used in Secure state, access DTCCR\_SEC instead of DTCCR.

## 17.6 Examples of DTC Usage

### 17.6.1 Normal Transfer

This section provides an example of DTC usage and its application when receiving 128 bytes of data from an SCI.

#### (1) Transfer information settings

In the MRA register, select a fixed source address (MRA.SM[1:0] = 00b), normal transfer mode (MRA.MD[1:0] = 00b), and byte-sized transfer (MRA.SZ[1:0] = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 10b) and single data transfer by a single interrupt (MRB.CHNE = 0 and MRB.DISEL = 0). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the SRAM area for data storage in the DAR register, and 128 (0x0080) in the CRA register. The CRB register can be set to any value.

#### (2) DTC vector table settings

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

#### (3) ICU settings and DTC module activation

Set the ICU.IELSRn.DTCE bit to 1 and set ICU.IELSRn.IELS[8:0] as the SCI interrupt. The interrupt must be enabled in the NVIC. Set the DTCST.DTCST bit to 1.

#### (4) SCI settings

Enable the SCIn\_RXI (n = 0 to 4, 9) interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, reception stops. To manage this, use settings that allow the CPU to accept receive error interrupts.

#### (5) DTC transfer

Each time a reception of 1 byte by the SCI is complete, an SCIn\_RXI interrupt is generated to activate the DTC. The DTC transfers the received byte from the RDR of the SCI to the SRAM, after which the DAR register is incremented and the CRA register is decremented.

#### (6) Interrupt handling

After 128 rounds of data transfer are complete and the value in the CRA register becomes 0, an SCIn\_RXI interrupt request is generated for the CPU. Complete the process in the handling routine for this interrupt.

### 17.6.2 Chain transfer

This section provides an example of chain transfer by the DTC and describes its use in the output of pulses by the General PWM Timer (GPT). You can use chain transfer to transfer PWM timer compare data and change the period of the PWM timer for the GPT.

For the first of the chain transfers, normal transfer mode is specified for transfer to the GPT32n.GTCCRC register (n = 0 to 5) or GPT16m.GTCCRC register (m = 10 to 13). For the second transfer, normal transfer mode is specified for transfer to the GPT32n.GTCCRE register (n = 0 to 5) or GPT16m.GTCCRE register (m = 10 to 13). For the third transfer of the chained transfer, normal transfer mode for transfer to the GPT32n.GTPBR register (n = 0 to 5) or GPT16m.GTPBR register (m = 10 to 13) is specified. This is because clearing of the activation source and generation of an interrupt on completion of the specified number of transfers are restricted to the third of the chain transfers, that is, transfer while MRB.CHNE = 0.

The following example shows how to use the counter overflow interrupt with the GPT320.GTPR register as an activating source for the DTC.

#### (1) First transfer information setting

Set up transfer to the GPT320.GTCCRC register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1 and MRB.CHNS = 0).

4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.GTCCRC register.
6. Set the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

## (2) Second transfer information setting

Set up for transfer to the GPT320.GTCCRE register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up chain transfer (MRB.CHNE = 1, MRB.CHNS = 0).
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.GTCCRE register.
6. Set the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

## (3) Third transfer information set

Set up transfer to the GPT320.GTPBR register.

1. In the MRA register, select incrementation of the source address (MRA.SM[1:0] = 10b).
2. Set the transfer to normal transfer mode (MRA.MD[1:0] = 00b) and word-sized transfer (MRA.SZ[1:0] = 10b).
3. In the MRB register, select the destination address as fixed (MRB.DM[1:0] = 00b) and set up single data transfer per interrupt (MRB.CHNE = 0, MRB.DISEL = 0). The MRB.DTS bit can be set to any value.
4. Set the SAR register to the first address of the data table.
5. Set the DAR register to the address of the GPT320.GTPBR register.
6. Set the CRA register to the size of the data table. The CRB register can be set to any value.

## (4) Transfer information assignment

Place the transfer information for use in the transfer to the GPT320.GTPBR immediately after the transfer control information for use in the GPT320.GTCCRC and GPT320.GTCCRE registers.

## (5) DTC vector table

In the DTC vector table, set the address where the transfer control information for use in transfer to the GPT320.GTCCRC and GPT320.GTCCRE registers starts.

## (6) ICU setting and DTC module activation

1. Set the ICU.IELSRn.DTCE bit associated with the GPT320 counter overflow interrupt.
2. Set the ICU.IELSRn.IELS[8:0] bits and specify the GPT320 counter overflow.
3. Set the DTCST.DTCST bit to 1.

## (7) GPT settings

1. Set the GPT320.GTIOR register so that the GTCCRA and GTCCRB registers operate as output compare registers.
2. Set the default PWM timer compare values in the GPT320.GTCCRA and GPT320.GTCCRB registers and the next PWM timer compare values in the GPT320.GTCCRC and GPT320.GTCCRE registers.
3. Set the default PWM timer period values in the GPT320.GTPR register and the next PWM timer period values in the GPT320.GTPBR register.
4. Set 1 to the output bit in PmnPFS.PDR, and set 00011b to the Peripheral Select bits in PmnPFS.PSEL[4:0].

## (8) GPT activation

Set the GPT320.GTSTR.CSTRT bits to 1 to start the GPT320.GTCNT counter.



### (9) DTC transfer

Each time a GPT320 counter overflow is generated with the GPT320.GTPR register, the next PWM timer compare values are transferred to the GPT320.GTCCRC and GPT320.GTCCRE registers. The setting for the next PWM timer period is transferred to the GPT320.GTPBR register.

### (10) Interrupt handling

After the specified rounds of data transfer are complete, for example when the value in the CRA register for GPT transfer becomes 0, a GPT320 counter overflow interrupt request is issued for the CPU. Complete the process for this interrupt in the handling routine.

## 17.6.3 Chain Transfer when Counter = 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second transfer. Chain transfer enables transfers to be repeated 256 times or more.

The following procedure shows an example of configuring a 1-KB input buffer, where the input buffer is set so that its lower address starts with 0x00. [Figure 17.14](#) shows a chain transfer when the counter = 0.

1. Set the normal transfer mode to input data for the first data transfer. Set the following:
  - (a) Transfer source address = fixed.
  - (b) CRA register = 0x0200 (512) times.
  - (c) MRB.CHNE bit = 1 (chain transfer is enabled).
  - (d) MRB.CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0).
  - (e) MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
2. Prepare the upper 8-bit address of the start address at every 512 times of the transfer destination address for the first data transfer in different area such as the flash. For example, when setting the input buffer to 0x8000 to 0x83FF, prepare 0x82 and 0x80.
3. For the second data transfer:
  - (a) Set the repeat transfer mode (with transfer source and destination address = fixed.) to reset the transfer counter of the first data transfer.
  - (b) Specify the CRA register in the first transfer information area for the transfer destination.
  - (c) Set the MRB.CHNE bit = 1 (chain transfer is enabled).
  - (d) Set the MRB.CHNS bit = 0 (select continuous chain transfer).
  - (e) Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
  - (f) CRA register = 0x0101 (The transfer count is 1).
4. For the third data transfer:
  - (a) Set the repeat transfer mode (with the source as the repeat area) to reset the transfer destination address of the first data transfer.
  - (b) Specify the upper 8 bits of the DAR register in the first transfer information area for the transfer destination.
  - (c) Set the MRB.CHNE bit = 0 (chain transfer is disabled).
  - (d) Set the MRB.DISEL bit = 0 (an interrupt request to the CPU is generated when the specified data transfer completes).
  - (e) When setting the input buffer to 0x8000 to 0x83FF, also set the transfer counter to 2.
5. The first data transfer is performed by an interrupt 512 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the transfer counter of the first data transfer to 0x0200. The lower 8 bits of the transfer destination address and the transfer counter of the first data transfer becomes 0x0200.
6. The second data transfer is performed by an interrupt 1 times. When the transfer counter of the first data transfer becomes 0, the third data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to

- 0x82. The lower 8 bits of the transfer destination address becomes 0x00 and the transfer counter of the first data transfer becomes 0x0200.
7. In succession, the first data transfer is performed by an interrupt 512 times as specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the transfer counter of the first data transfer to 0x0200. The lower 8 bits of the transfer destination address and the transfer counter of the first data transfer becomes 0x0200.
  8. The second data transfer is performed by an interrupt 1 times. When the transfer counter of the first data transfer becomes 0, the third data transfer starts. Set the upper 8 bits of the transfer destination address of the first data transfer to 0x80. The lower 8 bits of the transfer destination address becomes 0x00 and the transfer counter of the first data transfer becomes 0x0200.
  9. Steps 5 to 8 are repeated indefinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

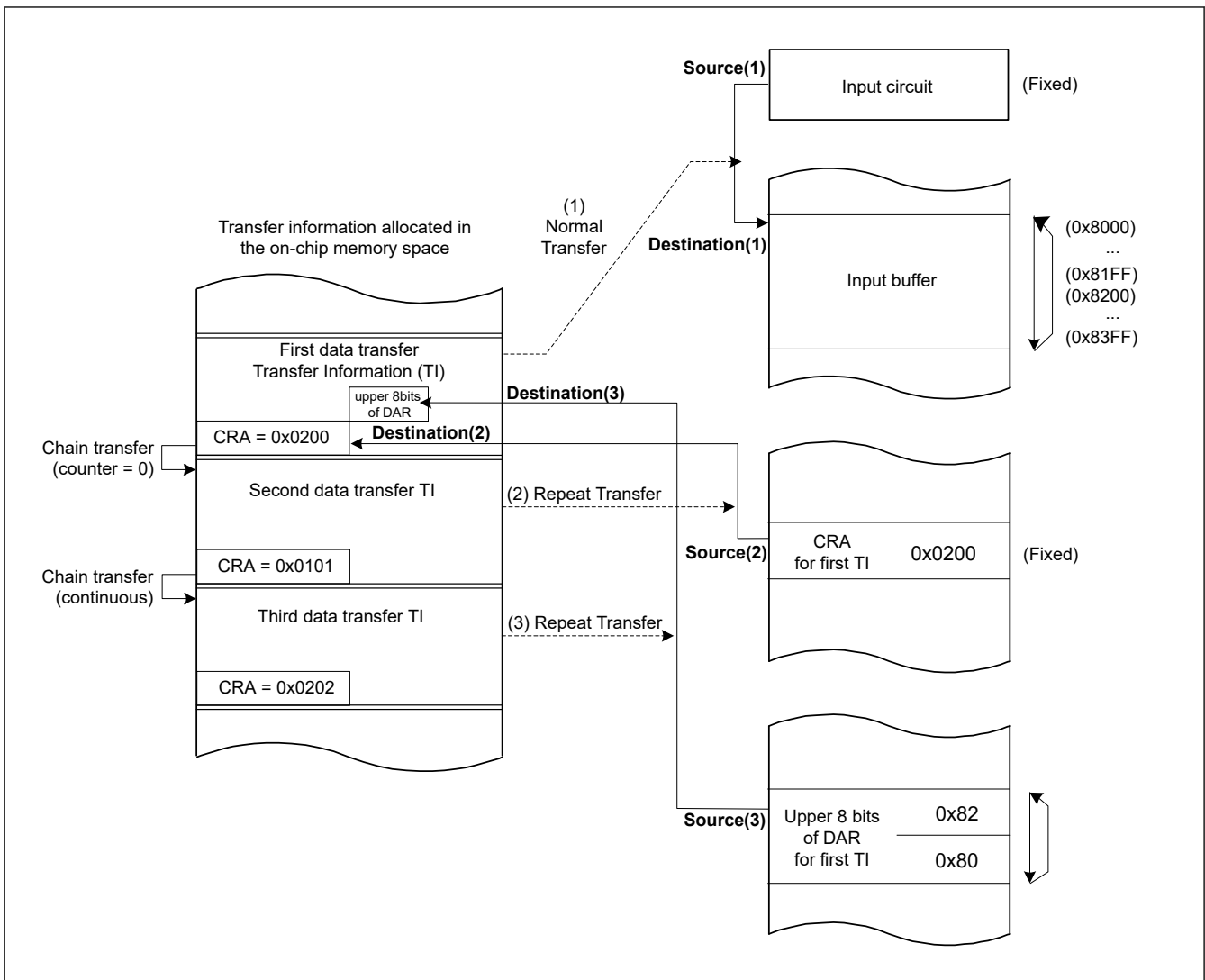


Figure 17.14 Chain transfer when counter = 0

### 17.7 Processing on DTC Transfer Error

If the access error occurs during DTC transfer, the DTC immediately stops access during transfer. DTC is stopped only for the vector number that caused the error. Therefore, inform the vector number that caused the error to the ICU and cleared the corresponding ICU setting (IELSR). After that, if there is a request other than the vector number which caused the error, it will be re-arbitration as it is.

The error response is informed to ICU when the transfer error occurs. ICU clears the ICU.IELSRn of the corresponding vector number which caused the transfer error. Write back to SRAM is not performed.

## 17.8 Interrupt

### 17.8.1 Interrupt Request of Transfer End

When the DTC completes data transfer of the specified count or when data transfer with MRB.DISEL set to 1 is complete, a DTC activation source generates an interrupt to the CPU. Two types of interrupt are available: interrupts triggered by a DTC activation (per channel) and an interrupt triggered by the event signal DTC0\_COMPLETE (common to all channels). Interrupts to the CPU are controlled according to the settings in the NVIC and the ICU.IELSRn.IELS[8:0] bits. See [section 13, Interrupt Controller Unit \(ICU\)](#). The DTC prioritizes activation sources by granting the smaller interrupt vector numbers higher priority. The priority of interrupts to the CPU is determined by the NVIC priority.

When the transfer destination is the external bus, an interrupt request is generated upon completion of data write to the write buffer not to the actual transfer destination.

### 17.8.2 Interrupt Request of Transfer Error

The error response detection interrupt request (DMA0\_TRANSERR) is generated from the DMAC/DTC when the transfer error is detected during DTC transfer.

The types of interrupts that occur when the DTC transfer error occurs are listed in [Table 17.9](#). The [Table 17.9](#) also shows error information stored when a transfer error occurs.

**Table 17.9 Interrupt and error information due to DTC transfer error cause**

Transfer error factor	NMI/RESET*1 Request	Interrupt Request	Bus Error Status	Error Address Error R/W	Error Channel Information
MSAU Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. MSERRSTAT	BUS.BMSA4ERRADD BUS.BMSA4ERRRW	DTC.DTEVR
Master MPU Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. MMERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DTC.DTEVR
Illegal Access Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. ILERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DTC.DTEVR
Slave Bus Error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. SLERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DTC.DTEVR
TrustZone Filter error	ICU.NMISR.BUSST	DMA0_TRANSERR*1	BUS.BUS4ERRSTAT. STERRSTAT	BUS.BUS4ERRADD BUS.BUS4ERRRW	DTC.DTEVR

Note 1. If ICU.NMIER.BUSEN is enabled with DMA0\_TRANSERR set in ICU.IELSR, NMI and an interrupt will occur due to a transfer error caused by DMA. It is able to be generated only NMI by not setting DMA0\_TRANSERR to ICU.IELSR.

## 17.9 Event Link

The DTC can produce an event link request on completion of one transfer request.

### 17.10 Low Power Consumption Function

Before transitioning to the module-stop state, Software Standby mode or Deep Software Standby mode, set the DTCST.DTCST bit to 0, and then perform the operations described in the following sections.

#### (1) Module-stop function

Writing 1 to the MSTPCRA.MSTPA22 bit enables the module-stop function of the DTC. If a DTC transfer is in progress when 1 is written to the MSTPCRA.MSTPA22 bit, the transition to the module-stop state proceeds after the DTC transfer ends. While the MSTPCRA.MSTPA22 bit is 1, accessing the DTC registers is prohibited. Writing 0 to the MSTPCRA.MSTPA22 bit releases the DTC from the module-stop state.

#### (2) Software Standby mode and Deep Software Standby mode

Use the settings described in [section 10.6.3.1. Transition to Software Standby Mode](#) or [section 10.6.4.1. Transition to Deep Software Standby Mode](#).

If DTC transfer operations are in progress when the WFI instruction is executed, the transition to Software Standby mode or Deep Software Standby mode follows the completion of the DTC transfer.

### (3) Notes on Low Power Consumption Function

For the WFI instruction and the register setting procedure, see [section 10, Low Power Modes](#).

To perform a DTC transfer after returning from a low power mode.

To use a request that is generated in Software Standby mode as an interrupt request to the CPU but not as a DTC activation request, specify the CPU as the interrupt request destination as described in [section 13.5.5. Selecting Interrupt Request Destinations](#), then execute the WFI instruction.

## 17.11 Usage Notes

### 17.11.1 Transfer Information Start Address

You must set multiples of 4 for the transfer information start addresses in the vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

### 17.11.2 Restriction for OSPI

There is a restriction to using OSPI, see [section 35.3.8.5. Restriction in 8D-8D-8D profile 1.0 format](#).

## 18. Event Link Controller (ELC)

### 18.1 Overview

The Event Link Controller (ELC) uses the event requests generated by various peripheral modules as source signals to connect them to different modules, allowing direct link between the modules without CPU intervention.

[Table 18.1](#) lists the ELC specifications, and [Figure 18.1](#) shows a block diagram.

**Table 18.1 ELC Specifications**

Item	Description
Event link function	196 types of event signals can be directly connected to modules. The ELC generates the ELC event signal, and events that activate the DTC.
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set for each registers

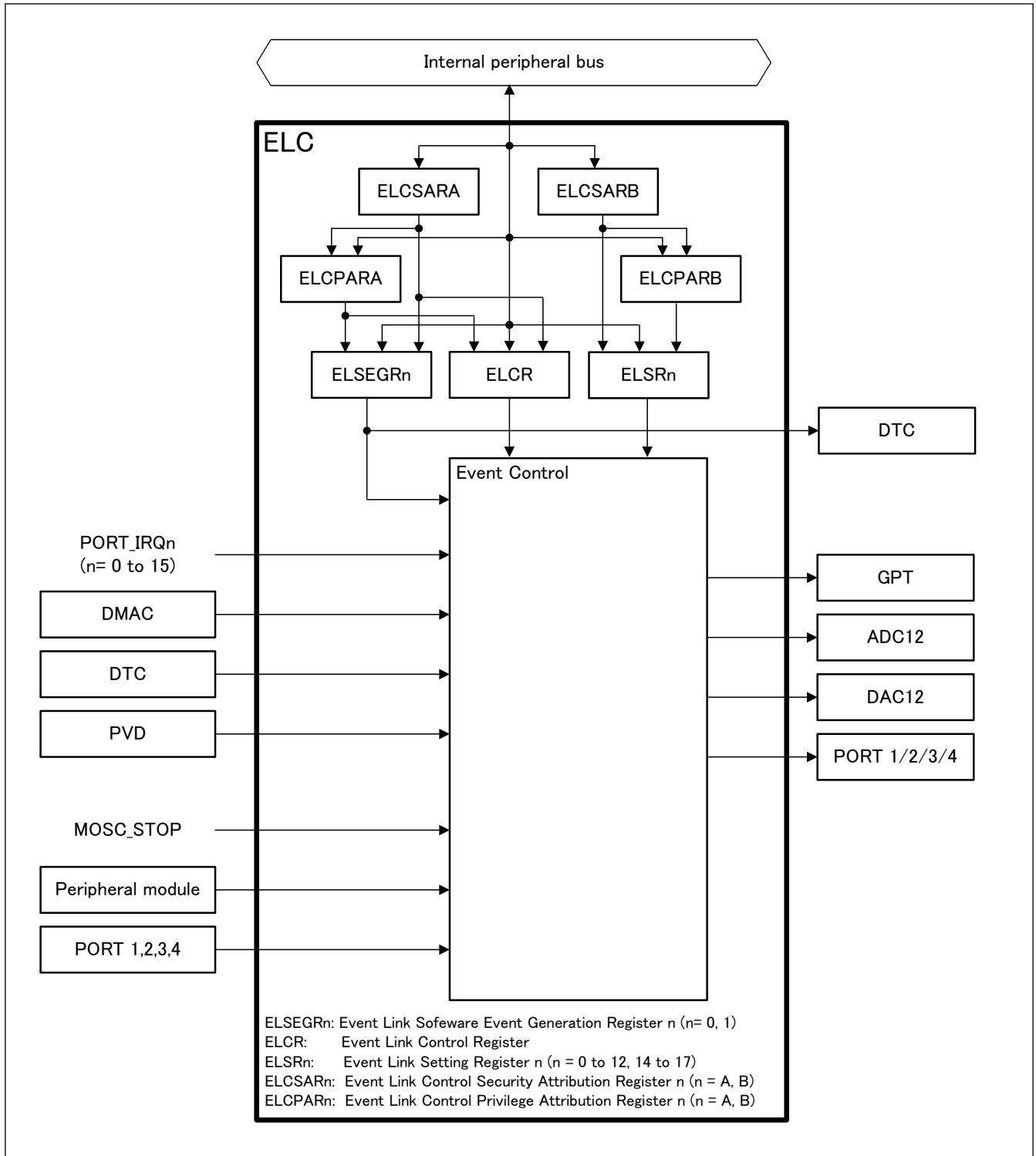


Figure 18.1 ELC block diagram

## 18.2 Register Descriptions

### 18.2.1 ELCR : Event Link Controller Register

Base address: ELC = 0x4020\_1000  
ELC\_NS = 0x5020\_1000

Offset address: 0x00

Bit position: 7 6 5 4 3 2 1 0

Bit field:	ELCON	—	—	—	—	—	—	—
------------	-------	---	---	---	---	---	---	---

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	ELCON	All Event Link Enable 0: ELC function is disabled. 1: ELC function is enabled.	R/W

Note: S-TYPE-3, P-TYPE-3

The ELCR register controls the ELC operation.

### 18.2.2 ELSEGRn : Event Link Software Event Generation Register n (n = 0, 1)

Base address: ELC = 0x4020\_1000  
ELC\_NS = 0x5020\_1000

Offset address: 0x04 + 0x04 × n

Bit position: 7 6 5 4 3 2 1 0

Bit field:	WI	WE	—	—	—	—	—	SEG
------------	----	----	---	---	---	---	---	-----

Value after reset: 1 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	SEG	Software Event Generation 0: Normal operation 1: Software event is generated.	W
5:1	—	These bits are read as 0. The write value should be 0.	R/W
6	WE	SEG Bit Write Enable 0: Write to SEG bit disabled. 1: Write to SEG bit enabled.	R/W
7	WI	ELSEGR Register Write Disable 0: Write to ELSEGR register enabled. 1: Write to ELSEGR register disabled.	W

Note: S-TYPE-3, P-TYPE-3

#### SEG bit (Software Event Generation)

When 1 is written to the SEG bit while the WE bit is 1, a software event is generated. This bit is read as 0. Even when 1 is written to this bit, data is not stored. The WE bit must be set to 1 before writing to this bit.

A software event can trigger a linked DTC event.

#### WE bit (SEG Bit Write Enable)

The SEG bit can only be written to when the WE bit is 1. Clear the WI bit to 0 before writing to this bit.

[Setting condition]

- If 1 is written to this bit while the WI bit is 0, this bit becomes 1.

[Clearing condition]

- If 0 is written to this bit while the WI bit is 0, this bit becomes 0.

**WI bit (ELSEGR Register Write Disable)**

The ELSEGR register can only be written to when the write value to the WI bit is 0. This bit is read as 1. Before setting the WE or SEG bit, the WI bit must be set to 0.

**18.2.3 ELSRn : Event Link Setting Register n (n = 0 to 12, 14 to 17)**

Base address: ELC = 0x4020\_1000  
 ELC\_NS = 0x5020\_1000

Offset address: 0x20 + 0x04 × n



Bit	Symbol	Function	R/W
8:0	ELS[8:0]	Event Link Select 0x000: Event output disabled for the associated peripheral module 0x001: Number setting for the event signal to be linked ⋮ 0x1BA: Number setting for the event signal to be linked Others: Settings prohibited	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: P-TYPE-3

Note: This register has different secure access permission depending on the setting of Trusted Event Route Control Register (TEVTRCR). If the security attribution is configured as Secure,

- Secure access is allowed.
- Non-secure write access is ignored and Non-secure read access is read as 0, TrustZone access error is generated.

If the security attribution is configured as Non-secure and the trusted event route is disabled,

- Secure write access is ignored and Secure read access is read as 0, TrustZone access error is generated.
- Non-secure access is allowed.

If the security attribution is configured as Non-secure and the trusted event route is enabled,

- Secure access is allowed.
- Non-secure write access is ignored and Non-secure read access is allowed.
- TrustZone access error is not generated.

The register specifies an event signal to be linked to each peripheral module. Table 18.2 shows the association between the register and the peripheral modules. Table 18.3 shows the association between the event signal names set in the register and the signal numbers.

**Table 18.2 Association between the registers and peripheral functions (1 of 2)**

Register name	Peripheral function (module)	Event name
ELSR0	GPT (A)	ELC_GPTA
ELSR1	GPT (B)	ELC_GPTB
ELSR2	GPT (C)	ELC_GPTC
ELSR3	GPT (D)	ELC_GPTD
ELSR4	GPT (E)	ELC_GPTE
ELSR5	GPT (F)	ELC_GPTF
ELSR6	GPT (G)	ELC_GPTG
ELSR7	GPT (H)	ELC_GPTH



**Table 18.2 Association between the registers and peripheral functions (2 of 2)**

Register name	Peripheral function (module)	Event name
ELSR8	ADC12A0	ELC_AD00
ELSR9	ADC12B0	ELC_AD01
ELSR10	ADC12A1	ELC_AD10
ELSR11	ADC12B1	ELC_AD11
ELSR12	DAC12 channel 0	ELC_DA0
ELSR14	PORT1	ELC_PORT1
ELSR15	PORT2	ELC_PORT2
ELSR16	PORT3	ELC_PORT3
ELSR17	PORT4	ELC_PORT4

**Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (1 of 6)**

Event number	Interrupt request source	Name	Description
0x001	Port	PORT_IRQ0*1	External pin interrupt 0
0x002		PORT_IRQ1*1	External pin interrupt 1
0x003		PORT_IRQ2*1	External pin interrupt 2
0x004		PORT_IRQ3*1	External pin interrupt 3
0x005		PORT_IRQ4*1	External pin interrupt 4
0x006		PORT_IRQ5*1	External pin interrupt 5
0x007		PORT_IRQ6*1	External pin interrupt 6
0x008		PORT_IRQ7*1	External pin interrupt 7
0x009		PORT_IRQ8*1	External pin interrupt 8
0x00A		PORT_IRQ9*1	External pin interrupt 9
0x00B		PORT_IRQ10*1	External pin interrupt 10
0x00C		PORT_IRQ11*1	External pin interrupt 11
0x00D		PORT_IRQ12*1	External pin interrupt 12
0x00E		PORT_IRQ13*1	External pin interrupt 13
0x00F		PORT_IRQ14*1	External pin interrupt 14
0x010		PORT_IRQ15*1	External pin interrupt 15
0x011	DMAC0	DMAC00_INT	DMAC0 transfer end 0
0x012		DMAC01_INT	DMAC0 transfer end 1
0x013		DMAC02_INT	DMAC0 transfer end 2
0x014		DMAC03_INT	DMAC0 transfer end 3
0x015		DMAC04_INT	DMAC0 transfer end 4
0x016		DMAC05_INT	DMAC0 transfer end 5
0x017		DMAC06_INT	DMAC0 transfer end 6
0x018		DMAC07_INT	DMAC0 transfer end 7
0x021	DTC0	DTC0_DTCEND	DTC0 transfer end
0x038	PVD	PVD_PVD1	Voltage monitor 1 interrupt
0x039		PVD_PVD2	Voltage monitor 2 interrupt
0x03E	MOSC	MOSC_STOP	Mail Clock oscillation stop

**Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (2 of 6)**

Event number	Interrupt request source	Name	Description
0x040	ULPT0	ULPT0_ULPTI	Underflow
0x041		ULPT0_ULPTCMAI	Compare match A
0x042		ULPT0_ULPTCMBI	Compare match B
0x043	ULPT1	ULPT1_ULPTI	Underflow
0x044		ULPT1_ULPTCMAI	Compare match A
0x045		ULPT1_ULPTCMBI	Compare match B
0x046	AGT0	AGT0_AGTI	AGT interrupt
0x047		AGT0_AGTCMAI	Compare match A
0x048		AGT0_AGTCMBI	Compare match B
0x049	AGT1	AGT1_AGTI	AGT interrupt
0x04A		AGT1_AGTCMAI	Compare match A
0x04B		AGT1_AGTCMBI	Compare match B
0x052	IWDT	IWDT_NMIUNDF	IWDT underflow
0x053	WDT	WDT0_NMIUNDF	WDT underflow
0x056	RTC	RTC_PRD	Periodic interrupt
0x05C	IIC0	IIC0_RXI	Receive data full
0x05D		IIC0_TXI	Transmit data empty
0x05E		IIC0_TEI	Transmit end
0x05F		IIC0_EEI	Transfer error
0x061	IIC1	IIC1_RXI	Receive data full
0x062		IIC1_TXI	Transmit data empty
0x063		IIC1_TEI	Transmit end
0x064		IIC1_EEI	Transfer error
0x07B	ACMPHS	ACMP_HS0*1	Comparator interrupt 0
0x07C		ACMP_HS1*1	Comparator interrupt 1
0x083	ELC	ELC_SWEVT0	Software event 0
0x084		ELC_SWEVT1	Software event 1
0x088	I/O Port	IOPORT_GROUP1	Port 1 event
0x089		IOPORT_GROUP2	Port 2 event
0x08A		IOPORT_GROUP3	Port 3 event
0x08B		IOPORT_GROUP4	Port 4 event
0x0A1	GPT0	GPT0_CCMPA	Compare match A
0x0A2		GPT0_CCMPB	Compare match B
0x0A3		GPT0_CMPC	Compare match C
0x0A4		GPT0_CMPD	Compare match D
0x0A5		GPT0_CMPE	Compare match E
0x0A6		GPT0_CMPF	Compare match F
0x0A7		GPT0_OVF	Overflow
0x0A8		GPT0_UDF	Underflow
0x0A9		GPT0_PC	Cycle count function end

**Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (3 of 6)**

Event number	Interrupt request source	Name	Description
0x0AA	GPT1	GPT1_CCMPA	Compare match A
0x0AB		GPT1_CCMPB	Compare match B
0x0AC		GPT1_CMPC	Compare match C
0x0AD		GPT1_CMPD	Compare match D
0x0AE		GPT1_CMPE	Compare match E
0x0AF		GPT1_CMPF	Compare match F
0x0B0		GPT1_OVF	Overflow
0x0B1		GPT1_UDF	Underflow
0x0B2		GPT1_PC	Cycle count function end
0x0B3		GPT2	GPT2_CCMPA
0x0B4	GPT2_CCMPB		Compare match B
0x0B5	GPT2_CMPC		Compare match C
0x0B6	GPT2_CMPD		Compare match D
0x0B7	GPT2_CMPE		Compare match E
0x0B8	GPT2_CMPF		Compare match F
0x0B9	GPT2_OVF		Overflow
0x0BA	GPT2_UDF		Underflow
0x0BB	GPT2_PC		Cycle count function end
0x0BC	GPT3		GPT3_CCMPA
0x0BD		GPT3_CCMPB	Compare match B
0x0BE		GPT3_CMPC	Compare match C
0x0BF		GPT3_CMPD	Compare match D
0x0C0		GPT3_CMPE	Compare match E
0x0C1		GPT3_CMPF	Compare match F
0x0C2		GPT3_OVF	Overflow
0x0C3		GPT3_UDF	Underflow
0x0C4		GPT3_PC	Cycle count function end
0x0C5		GPT4	GPT4_CCMPA
0x0C6	GPT4_CCMPB		Compare match B
0x0C7	GPT4_CMPC		Compare match C
0x0C8	GPT4_CMPD		Compare match D
0x0C9	GPT4_CMPE		Compare match E
0x0CA	GPT4_CMPF		Compare match F
0x0CB	GPT4_OVF		Overflow
0x0CC	GPT4_UDF		Underflow

**Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (4 of 6)**

Event number	Interrupt request source	Name	Description
0x0CE	GPT5	GPT5_CCMPA	Compare match A
0x0CF		GPT5_CCMPB	Compare match B
0x0D0		GPT5_CMPC	Compare match C
0x0D1		GPT5_CMPD	Compare match D
0x0D2		GPT5_CMPE	Compare match E
0x0D3		GPT5_CMPF	Compare match F
0x0D4		GPT5_OVF	Overflow
0x0D5		GPT5_UDF	Underflow
0x0FB	GPT10	GPT10_CCMPA	Compare match A
0x0FC		GPT10_CCMPB	Compare match B
0x0FD		GPT10_CMPC	Compare match C
0x0FE		GPT10_CMPD	Compare match D
0x0FF		GPT10_CMPE	Compare match E
0x100		GPT10_CMPF	Compare match F
0x101		GPT10_OVF	Overflow
0x102		GPT10_UDF	Underflow
0x103	GPT10_PC	Cycle count function end	
0x104	GPT11	GPT11_CCMPA	Compare match A
0x105		GPT11_CCMPB	Compare match B
0x106		GPT11_CMPC	Compare match C
0x107		GPT11_CMPD	Compare match D
0x108		GPT11_CMPE	Compare match E
0x109		GPT11_CMPF	Compare match F
0x10A		GPT11_OVF	Overflow
0x10B		GPT11_UDF	Underflow
0x10D	GPT12	GPT12_CCMPA	Compare match A
0x10E		GPT12_CCMPB	Compare match B
0x10F		GPT12_CMPC	Compare match C
0x110		GPT12_CMPD	Compare match D
0x111		GPT12_CMPE	Compare match E
0x112		GPT12_CMPF	Compare match F
0x113		GPT12_OVF	Overflow
0x114		GPT12_UDF	Underflow
0x116	GPT13	GPT13_CCMPA	Compare match A
0x117		GPT13_CCMPB	Compare match B
0x118		GPT13_CMPC	Compare match C
0x119		GPT13_CMPD	Compare match D
0x11A		GPT13_CMPE	Compare match E
0x11B		GPT13_CMPF	Compare match F
0x11C		GPT13_OVF	Overflow
0x11D		GPT13_UDF	Underflow

**Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (5 of 6)**

Event number	Interrupt request source	Name	Description
0x124	SCI0	SCI0_RXI*2	Receive data full
0x125		SCI0_TXI*2	Transmit data empty
0x126		SCI0_TEI*2	Transmit end
0x127		SCI0_ERI	Receive error
0x128		SCI0_AED	Effective edge detection
0x12A		SCI0_AM	Address match event
0x12B		SCI1	SCI1_RXI*2
0x12C	SCI1_TXI*2		Transmit data empty
0x12D	SCI1_TEI*2		Transmit end
0x12E	SCI1_ERI		Receive error
0x12F	SCI1_AED		Effective edge detection
0x131	SCI1_AM		Address match event
0x132	SCI2		SCI2_RXI*2
0x133		SCI2_TXI*2	Transmit data empty
0x134		SCI2_TEI*2	Transmit end
0x135		SCI2_ERI	Receive error
0x138		SCI2_AM	Address match event
0x139		SCI3	SCI3_RXI*2
0x13A	SCI3_TXI*2		Transmit data empty
0x13B	SCI3_TEI*2		Transmit end
0x13C	SCI3_ERI		Receive error
0x13F	SCI3_AM		Address match event
0x140	SCI4		SCI4_RXI*2
0x141		SCI4_TXI*2	Transmit data empty
0x142		SCI4_TEI*2	Transmit end
0x143		SCI4_ERI	Receive error
0x146		SCI4_AM	Address match event
0x163		SCI9	SCI9_RXI*2
0x164	SCI9_TXI*2		Transmit data empty
0x165	SCI9_TEI*2		Transmit end
0x166	SCI9_ERI		Receive error
0x169	SCI9_AM		Address match event
0x178	SPI0		SPI0_SPRI
0x179		SPI0_SPTI	Transmit buffer empty
0x17A		SPI0_SPII	Idle
0x17B		SPI0_SPEI	Error
0x17C		SPI0_SPCEND	Communication complete event

**Table 18.3 Association between event signal names set in .ELS[8:0] bits and signal numbers (6 of 6)**

Event number	Interrupt request source	Name	Description
0x17D	SPI1	SPI1_SPRI	Receive buffer full
0x17E		SPI1_SPTI	Transmit buffer empty
0x17F		SPI1_SPII	Idle
0x180		SPI1_SPEI	Error
0x181		SPI1_SPCEND	Transmission complete event
0x1AE	ADC12	ADC120_ADI	A/D scan end interrupt
0x1AF		ADC120_GBADI	A/D scan end interrupt for group B
0x1B2		ADC120_WCMPPM	Compare match
0x1B3		ADC120_WCMPUM	Compare mismatch
0x1B4		ADC121_ADI	A/D scan end interrupt
0x1B5		ADC121_GBADI	A/D scan end interrupt for group B
0x1B8		ADC121_WCMPPM	Compare match
0x1B9		ADC121_WCMPUM	Compare mismatch
0x1BA	DOC	DOC_DOPCI	Data operation circuit interrupt

Note 1. Only pulse (edge detection) is supported.  
 Note 2. This event is not supported in FIFO mode.

### 18.2.4 ELCSARA : Event Link Controller Security Attribution Register A

Base ELC = 0x4020\_1000  
 address: ELC\_NS = 0x5020\_1000

Offset address: 0xE0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	ELSE GR1	ELSE GR0	ELCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ELCR	Event Link Controller Register Security Attribution Target register: ELCR 0: Secure 1: Non-secure	R/W
1	ELSEGR0	Event Link Software Event Generation Register 0 Security Attribution 0: Secure 1: Non-secure	R/W
2	ELSEGR1	Event Link Software Event Generation Register 1 Security Attribution 0: Secure 1: Non-secure	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1  
 Note: This register is write-protected by PRSCR\_S register.

This register specifies the security attribution for the ELCR, ELSEGR0 and ELSEGR1 registers.

### 18.2.5 ELCSARB : Event Link Controller Security Attribution Register B

Base address: ELC = 0x4020\_1000  
 ELC\_NS = 0x5020\_1000

Offset address: 0xE4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ELSR[17:16]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ELSR[15:14]		—	ELSR[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	ELSR[12:0]	Event Link Setting Register n Security Attribution Target register: ELSRn (n = 0 to 12) 0: Secure 1: Non-secure	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
17:14	ELSR[17:14]	Event Link Setting Register n Security Attribution Target register: ELSRn (n = 14 to 17) 0: Secure 1: Non-secure	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

Note: This register is write-protected by PRCR\_S register.

This register specifies the security attribution for the Register ELSRn (n = 0 to 12, 14 to 17).

### 18.2.6 ELCPARA : Event Link Controller Privilege Attribution Register A

Base address: ELC = 0x4020\_1000  
 ELC\_NS = 0x5020\_1000

Offset address: 0xF0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	ELSE GR1	ELSE GR0	ELCR
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	ELCR	Event Link Controller Register Privilege Attribution Target register: ELCR 0: Privileged 1: Unprivileged	R/W
1	ELSEGR0	Event Link Software Event Generation Register 0 Privilege Attribution Target register: ELSEGR0 0: Privileged 1: Unprivileged	R/W

Bit	Symbol	Function	R/W
2	ELSEGR1	Event Link Software Event Generation Register 1 Privilege Attribution Target register: ELSEGR1 0: Privileged 1: Unprivileged	R/W
31:3	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1

Note: This register is write-protected by PRPCR\_S and PRPCR\_NS register.

This register specifies the privilege attribution for the Register ELCR, ELSEGR0 and ELSEGR1 registers.

## 18.2.7 ELCPARB : Event Link Controller Privilege Attribution Register B

Base address: ELC = 0x4020\_1000  
ELC\_NS = 0x5020\_1000

Offset address: 0xF4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ELSR[17:16]	
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ELSR[15:14]		—	ELSR[12:0]												
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
12:0	ELSR[12:0]	Event Link Setting Register n Privilege Attribution Target register: ELSRn (n = 0 to 12) 0: Privileged 1: Unprivileged	R/W
13	—	This bit is read as 1. The write value should be 1.	R/W
17:14	ELSR[17:14]	Event Link Setting Register n Privilege Attribution Target register: ELSRn (n = 14 to 17) 0: Privileged 1: Unprivileged	R/W
31:18	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1

Note: This register is write-protected by PRPCR\_S and PRPCR\_NS register.

This register specifies the privilege attribution for the Register ELSRn (n = 0 to 12, 14 to 17) registers.

## 18.3 Operation

### 18.3.1 Relation between Interrupt Handling and Event Linking

Event number for an event link is the same as that for the associated interrupt source. For information on generating event signals, see the explanation in the chapter for each event source module.

### 18.3.2 Linking Events

When an event occurs and that event is already set as a trigger in the Event Link Setting Register (ELSRn), the associated module is activated. The operation of the module must be set up in advance. Table 18.4 lists the operations of modules when an event occurs.



**Table 18.4** Module operations when event occurs

Module	Operations When Event is Input
GPT	<ul style="list-style-type: none"> <li>• Start counting</li> <li>• Stop counting</li> <li>• Clear counting</li> <li>• Up counting</li> <li>• Down counting</li> <li>• Input capture</li> </ul>
DAC12	Start D/A conversion
I/O Ports	<ul style="list-style-type: none"> <li>• Change pin output based on the EORR (reset) or EOSR (set)</li> <li>• Latch pin state to EIDR</li> <li>• The following ports can be used for the ELC: PORT 1 PORT 2 PORT 3 PORT 4</li> </ul>
ADC12	Start A/D conversion
DTC	Start DTC data transfer

### 18.3.3 Example of Procedure for Linking Events

To link events:

1. Set the operation of the module for which an event is to be linked.
2. Set the appropriate ELSRn.ELS[8:0] bits for the module to be linked.
3. Set the ELCR.ELCON bit to 1 to enable linkage of all events.
4. Configure the module from which an event is output and activate the module. The link between the two modules is now active.
5. To stop event linkage of modules individually, set 0 to the ELSRn.ELS[8:0] bit associated with the modules. To stop linkage of all the events, set the ELCR.ELCON bit to 0.

If event link output from the RTC is to be used, set the ELC after the RTC settings, for example, for initialization and time setting. Unintended events may be generated if RTC settings are made after the ELC settings.

If event link output from the PVD is to be used, set the ELC after setting the PVD. To disable the PVD, do so after setting 0x00 to the associated ELSRn register.

## 18.4 Usage Notes

### 18.4.1 Linking DMAC/DTC Transfer End Signals as Events

When linking the DMAC/DTC transfer end signals as events, do not set the same peripheral module as the DMAC/DTC transfer destination and event link destination. If set, the peripheral module might be started before DMAC/DTC transfer to the peripheral module is complete.

### 18.4.2 Setting Clocks

To link events, you must enable the ELC and the related modules. The modules cannot operate if the related modules are in the module-stop state or in low power mode in which the module is stopped (CPU Deep Sleep mode, Software Standby mode or Deep Software Standby mode). For more information, see [Table 18.3](#) and [section 10, Low Power Modes](#).

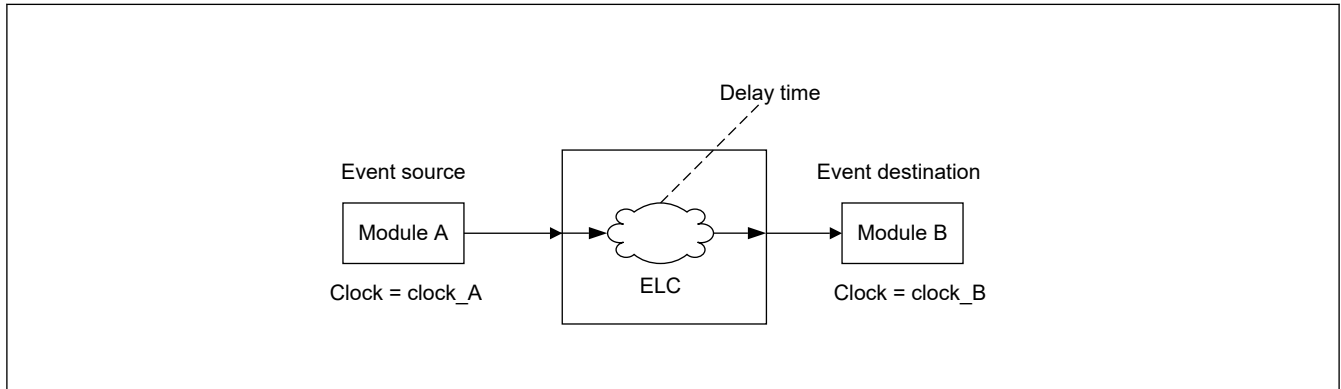
### 18.4.3 Module-Stop Function Setting

The Module Stop Control Register C (MSTPCRC) can enable or disable ELC operation. The ELC is initially stopped after reset. Releasing the module-stop state enables access to the registers. The ELCON bit must be set to 0 before disabling ELC operation using the Module Stop Control Register. For more information, see [Table 18.3](#) and [section 10, Low Power Modes](#).

### 18.4.4 ELC Delay Time

In [Figure 18.2](#), module A accesses module B through the ELC. There is a delay time in the ELC between module A and module B. [Table 18.5](#) shows the ELC delay time.

[Table 18.6](#) shows the clock domain of each module A. [Table 18.7](#) shows the clock domain of each module B.



**Figure 18.2 ELC delay time**

**Table 18.5 ELC delay time**

Clock domain*1	Clock frequency	ELC delay time
clock_A = clock_B	clock_A = clock_B	0 cycle
clock_A ≠ clock_B	clock_A = clock_B	1 cycle to 2 cycles
	clock_A > clock_B	1 cycle to 2 cycles of clock_B
	clock_A < clock_B	1 cycle to 2 cycles of clock_A

Note 1. When PORT\_IRQ, PVD, MOSC, DMAC, DTC, I/O Ports is selected as event source module, the ELC delay time is 2 cycles to 4 cycles of PCLKB.

**Table 18.6 Clock domain of each module A**

Module A	Clock domain of clock_A
PORT_IRQ, PVD, MOSC	ASYNC
DMAC, DTC, I/O Ports	ICLK
ULPT, AGT, IWDT, CWDT, IIC, ACMPHS, ELC, RTC	PCLKB
GPT	PCLKD
SCI, SPI, ADC12, DOC	PCLKA

**Table 18.7 Clock domain of each module B**

Module B	Clock domain of clock_B
ADC12, DAC12	PCLKA
DTC, I/O Ports	ICLK
GPT	PCLKD

## 19. I/O Ports

### 19.1 Overview

The I/O port pins operate as general I/O port pins, I/O pins for peripheral modules, interrupt input pins, analog I/O, port group function for the ELC, or bus control pins.

All pins except P209 ( as TDO of JTAG ports ) operate as input pins immediately after a reset, and pin functions are switched by register settings. The I/O ports and peripheral modules for each pin are specified in the associated registers.

Figure 19.1 shows a connection diagram for the I/O port registers. The configuration of the I/O ports differs for different packages. Table 19.1 lists the I/O port specifications by package, and Table 19.2 lists the port functions.

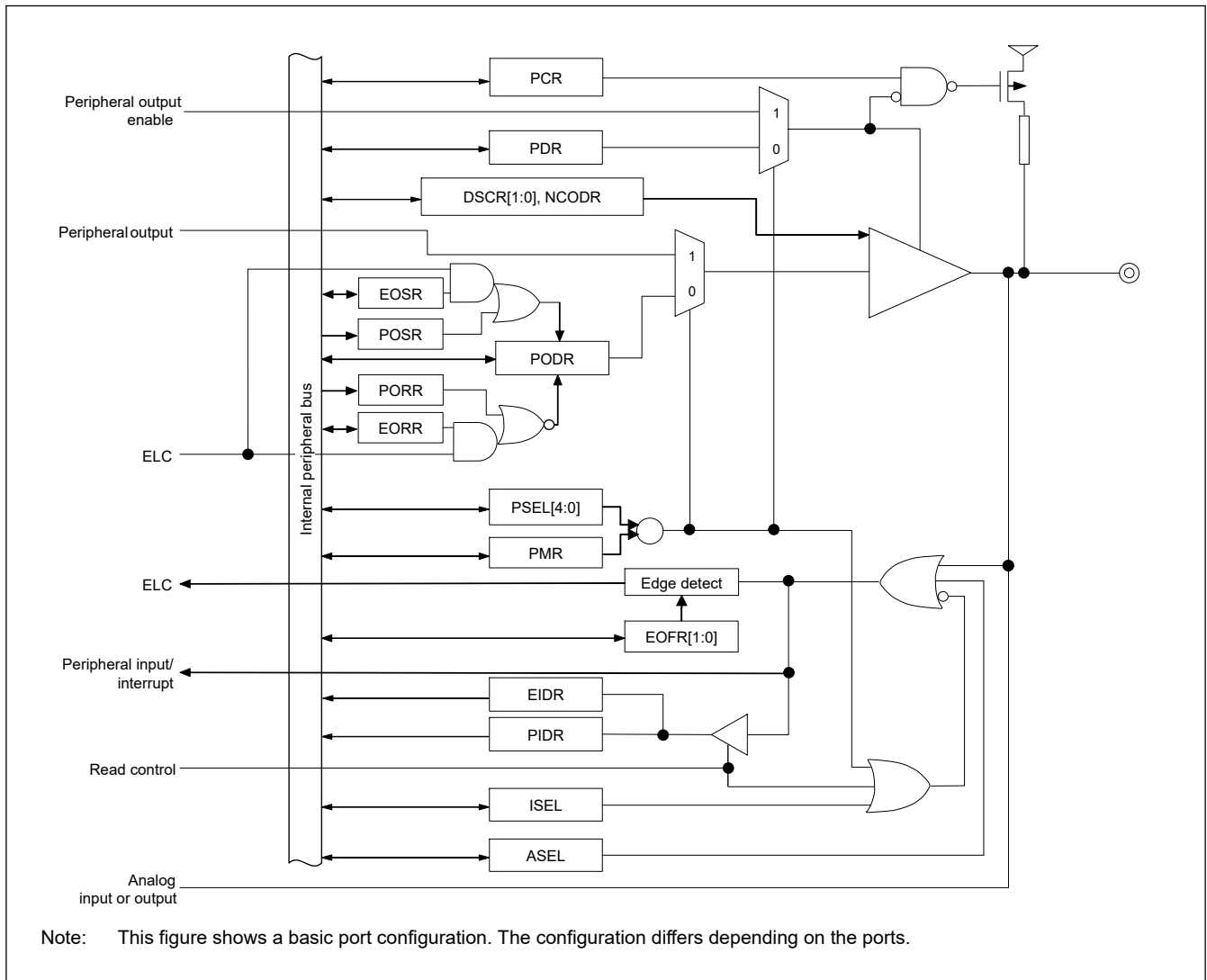


Figure 19.1 Connection diagram for I/O port registers

Table 19.1 I/O port specifications (1 of 2)

Port	Package		Package	
	144 pins	Number of pins	100 pins	Number of pins
PORT0	P000 to P010, P014, P015	13	P000 to P010, P014, P015	11
PORT1	P100 to P107, P112 to P115	12	P100 to P107, P112 to P115	12

Table 19.1 I/O port specifications (2 of 2)

Port	Package		Package	
	144 pins	Number of pins	100 pins	Number of pins
PORT2	P200 to P213	13	P200, P201, P205, P206, P208 to P213	10
PORT3	P300 to P313	16	P300 to P308	9
PORT4	P400 to P415	16	P400 to P415	16
PORT5	P511 to P513	3	—	0
PORT6	P600 to P605, P609 to P614	12	P600, P609, P610	3
PORT7	P700 to P705, P708 to P713	12	P708	1
PORT8	P800 to P806, P808, P809, P814, P815	11	P808, P809, P814, P815	9
PORT9	P905	1	—	0

Table 19.2 I/O port functions

Port	Port name	Input pull-up	Open-drain output	Drive capability switching	5Vtolerant	I/O	Power supply
PORT0	P000 to P010, P014, P015	✓	✓	Low	—	Input / Output	AVCC0
PORT1	P100 to P103	✓	✓	Low, middle, high, high-speed high-drive	—	Input / Output	VCC2
	P104 to P107	✓	✓	Low, middle, high	—	Input / Output	VCC2
	P112 to P115	✓	✓	Low, middle, high	—	Input / Output	VCC
PORT2	P200	✓	—	—	—	Input	VCC
	P201	✓	✓	Low	—	Input / Output	VCC
	P202 to P204, P207 to P213	✓	✓	Low, middle, high	—	Input / Output	VCC
	P205, P206	✓	✓	Low, middle, high	✓	Input / Output	VCC
PORT3	P300 to P303, P309 to P313	✓	✓	Low, middle, high	—	Input / Output	VCC
	P304 to P308	✓	✓	Low, middle, high, high-speed high-drive	—	Input / Output	VCC
PORT4	P400, P401, P405	✓	✓	Low, middle, high	—	Input / Output	VCC
	P402 to P404, P406 to P415	✓	✓	Low, middle, high	✓	Input / Output	VCC
PORT5	P513	✓	✓	Low, middle, high	—	Input / Output	VCC
	P511, P512	✓	✓	Low, middle, high	✓	Input / Output	VCC
PORT6	P600	✓	✓	Low, middle, high	—	Input / Output	VCC2
	P601 to P605, P609 to P614	✓	✓	Low, middle, high	—	Input / Output	VCC
PORT7	P700 to P705, P708	✓	✓	Low, middle, high	—	Input / Output	VCC
	P709 to P713	✓	✓	Low, middle, high	✓	Input / Output	VCC
PORT8	P800 to P804, P808, P809	✓	✓	Low, middle, high, high-speed high-drive	—	Input / Output	VCC2
	P805, P806, P814, P815	✓	✓	Low, middle, high	—	Input / Output	VCC
PORT9	P905	✓	✓	Low, middle, high	—	Input / Output	VCC

Note: ✓: Available

—: Setting prohibited

## 19.2 Register Descriptions

### 19.2.1 PCNTR1/PODR/PDR : Port Control Register 1

Base address: PORTm = 0x4040\_0000 + 0x0020 × m (m = 0 to 9)  
 PORTm\_NS = 0x5040\_0000 + 0x0020 × m (m = 0 to 9)

Offset address: 0x000 (PCNTR1/PDR)  
 0x002 (PODR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PODR 15	PODR 14	PODR 13	PODR 12	PODR 11	PODR 10	PODR 09	PODR 08	PODR 07	PODR 06	PODR 05	PODR 04	PODR 03	PODR 02	PODR 01	PODR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PDR1 5	PDR1 4	PDR1 3	PDR1 2	PDR11	PDR1 0	PDR0 9	PDR0 8	PDR0 7	PDR0 6	PDR0 5	PDR0 4	PDR0 3	PDR0 2	PDR0 1	PDR0 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PDR15 to PDR00	Pmn Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
31:16	PODR15 to PODR00	Pmn Output Data 0: Low output 1: High output	R/W

Note: S-TYPE-4, P-TYPE-5

Note: m = 0 to 9, n = 00 to 15

The Port Control Register 1 (PCNTR1/PODR/PDR) is a 32-bit or 16-bit read/write register that controls port direction and port output data. The PCNTR1 specifies the port direction and output data, and is accessed in 32-bit units. The PDRn (bits [15:0] in PCNTR1) and PODRn (bits [31:16] in PCNTR1) respectively, are accessed in 16-bit units.

#### PDRn bits (Pmn Direction)

The PDRn bits select the input or output direction for individual pins on the associated port when the pins are configured as general I/O pins. Each pin on port m is associated with a PORTm.PCNTR1.PDRn bit. The I/O direction can be specified in 1-bit unit. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0. In the case of input only ports, PDRn bits are reserved. See [section 19.1. Overview](#). The PDRn bit in the PORTm.PCNTR1 register serves the same function as the PDR bit in the PFS.PmnPFS register.

#### PODRn bits (Pmn Output Data)

The PODRn bits hold data to be output from the general I/O pins. Bits of non-existent port m are reserved. Reserved bits are read as 0. The write value should be 0. In the case of input only ports, PODRn bits are reserved. See [section 19.1. Overview](#). The PODRn bit in the PORTm.PCNTR1 register serves the same function as the PODR bit in the PFS.PmnPFS register.

### 19.2.2 PCNTR2/EIDR/PIDR : Port Control Register 2

Base address: PORTm = 0x4040\_0000 + 0x0020 × m (m = 0 to 9)  
 PORTm\_NS = 0x5040\_0000 + 0x0020 × m (m = 0 to 9)

Offset address: 0x004 (PCNTR2/PIDR)  
 0x006 (EIDR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EIDR1 5	EIDR1 4	EIDR1 3	EIDR1 2	EIDR1 1	EIDR1 0	EIDR0 9	EIDR0 8	EIDR0 7	EIDR0 6	EIDR0 5	EIDR0 4	EIDR0 3	EIDR0 2	EIDR0 1	EIDR0 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PIDR1 5	PIDR1 4	PIDR1 3	PIDR1 2	PIDR1 1	PIDR1 0	PIDR0 9	PIDR0 8	PIDR0 7	PIDR0 6	PIDR0 5	PIDR0 4	PIDR0 3	PIDR0 2	PIDR0 1	PIDR0 0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
15:0	PIDR15 to PIDR00	Pmn State 0: Low level 1: High level	R
31:16	EIDR15 to EIDR00 *2	Port Event Input Data*1 When an ELC_PORTx signal occurs 0: Low input 1: High input	R

Note: S-TYPE-4, P-TYPE-5

Note: m = 0 to 9, n = 00 to 15

Note 1. x = 1, 2, 3 or 4 for EIDR only

Note 2. Supported by ports 1, 2, 3 or 4.

The Port Control Register 2 (PCNTR2/EIDR/PIDR) allows read access to the Pmn state and the port event input data using 32-bit or 16-bit access.

The PCNTR2 represents the Pmn state and the port event input data, and is accessed in 32-bit units.

The PIDRn (bits [15:0] in PCNTR2) and EIDRn (bits [31:16] in PCNTR2) respectively, are accessed in 16-bit units. Bits associated with non-existent pins are reserved. Reserved bits are read as undefined.

#### PIDRn bits (Pmn State)

The PIDRn bits reflect the individual pin states of the port, regardless of the values set in PmnPFS.PMR and PORTm.PCNTR1.PDRn. The PIDRn bit in the PORTm.PCNTR2 register serves the same function as the PIDR bit in the PFS.PmnPFS register.

A pin state cannot be reflected in PIDRn when one of the following functions is enabled:

- RTC Time Capture input and tamper detection (RTCIC)
- Analog function (ASEL = 1)
- Trace interface (TCLK, TDATA<sub>n</sub> (n = 0 to 3), SWO)

#### EIDRn bits (Port Event Input Data)

The EIDRn bits latch a pin state when an ELC\_PORTx signal occurs. Pin states can only be input to EIDRn when PmnPFS.PMR and PORTm.PCNTR1.PDRn are 0. When the PmnPFS.ASEL bit is set to 1, the associated pin state is not reflected in EIDRn.

### 19.2.3 PCNTR3/PORR/POSR : Port Control Register 3

Base address: PORTm = 0x4040\_0000 + 0x0020 × m (m = 0 to 9)  
 PORTm\_NS = 0x5040\_0000 + 0x0020 × m (m = 0 to 9)

Offset address: 0x008 (PCNTR3/POSR)  
 0x00A (PORR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PORR 15	PORR 14	PORR 13	PORR 12	PORR 11	PORR 10	PORR 09	PORR 08	PORR 07	PORR 06	PORR 05	PORR 04	PORR 03	PORR 02	PORR 01	PORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	POSR 15	POSR 14	POSR 13	POSR 12	POSR 11	POSR 10	POSR 09	POSR 08	POSR 07	POSR 06	POSR 05	POSR 04	POSR 03	POSR 02	POSR 01	POSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	POSR15 to POSR00	Pmn Output Set 0: No effect on output 1: High output	W
31:16	PORR15 to PORR00	Pmn Output Reset 0: No effect on output 1: Low output	W

Note: S-TYPE-4, P-TYPE-5

Note: m = 0 to 9, n = 00 to 15

The Port Control Register 3 (PCNTR3/PORR/POSR) is a 32-bit or 16-bit write register that controls the setting or resetting of the port output data.

The PCNTR3 controls the setting or resetting of the port output data, and is accessed in 32-bit units.

The POSRn (bits [15:0] in PCNTR3) and the PORRn (bits [31:16] in PCNTR3) respectively, are accessed in 16-bit units.

#### POSRn bits (Pmn Output Set)

POSR changes PODR when set by a software write. For example, for P100, when PORT1.PCNTR3.POSR00 = 1, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. In the case of input only ports, POSRn bits are reserved. See [section 19.1. Overview](#).

#### PORRn bits (Pmn Output Reset)

PORR changes PODR when reset by a software write. For example, for P100, when PORT1.PCNTR3.PORR00 = 1, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. In the case of input only ports, PORRn bits are reserved. See [section 19.1. Overview](#).

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.

Note: PORRn and POSRn should not be set at the same time.

## 19.2.4 PCNTR4/EORR/EOSR : Port Control Register 4

Base address: PORTm = 0x4040\_0000 + 0x0020 × m (m = 1 to 4)  
 PORTm\_NS = 0x5040\_0000 + 0x0020 × m (m = 1 to 4)

Offset address: 0x00C (PCNTR4/EOSR)  
 0x00E (EORR)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EORR 15	EORR 14	EORR 13	EORR 12	EORR 11	EORR 10	EORR 09	EORR 08	EORR 07	EORR 06	EORR 05	EORR 04	EORR 03	EORR 02	EORR 01	EORR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EOSR 15	EOSR 14	EOSR 13	EOSR 12	EOSR 11	EOSR 10	EOSR 09	EOSR 08	EOSR 07	EOSR 06	EOSR 05	EOSR 04	EOSR 03	EOSR 02	EOSR 01	EOSR 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	EOSR15 to EOSR00	Pmn Event Output Set When an ELC_PORTx signal occurs 0: No effect on output 1: High output	R/W
31:16	EORR15 to EORR0	Pmn Event Output Reset When an ELC_PORTx signal occurs 0: No effect on output 1: Low output	R/W

Note: S-TYPE-4, P-TYPE-5

Note: m = 1 to 4, n = 00 to 15, x = 1 to 4

The Port Control Register 4 (PCNTR4/EORR/EOSR) is a 32-bit or 16-bit read/write register that controls the setting or resetting of the port output data by an event input from the ELC.

The PCNTR4 controls the setting or resetting of the port output data by an event input from the ELC, and is accessed in 32-bit units.

The EOSRn (bits [15:0] in PCNTR4) and EORRn (bits [31:16] in PCNTR4) respectively, are accessed in 16-bit units.

### EOSRn bits (Pmn Event Output Set)

EOSR changes PODR when set because an ELC\_PORTx signal occurs. For example, for P100 if PORT1.PCNTR4.EOSR00 is set to 1 when the ELC\_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 1. Bits associated with non-existent pins are reserved. The write value should always be 0. For input only ports, EOSRn bits are reserved. See [section 19.1. Overview](#).

### EORRn bits (Pmn Event Output Reset)

EORR changes PODR when reset because an ELC\_PORTx signal occurs. For example, for P100 if PORT1.PCNTR4.EORR00 = 1 when the ELC\_PORTx occurs, PORT1.PCNTR1.PODR00 outputs 0. Bits associated with non-existent pins are reserved. The write value should always be 0. For input only ports, EORRn bits are reserved. See [section 19.1. Overview](#).

Note: When EORRn or EOSRn is set, writing is prohibited to PODRn, PORRn, and POSRn.

Note: EORRn and EOSRn should not be set at the same time.



### 19.2.5 PmnPFS/PmnPFS\_HA/PmnPFS\_BY : Port mn Pin Function Select Register (m = 0 to 9, n = 00 to 15)

Base address: PFS = 0x4040\_0800  
PFS\_NS = 0x5040\_0800

Offset address: 0x000 + 0x040 × m + 0x004 × n (PmnPFS/PmnPFS\_HA/PmnPFS\_BY)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Bit field:	—	—	—	PSEL[4:0]				—	—	—	—	—	—	—	—	—	PMR
Value after reset:	0	0	0	0 <sup>*1</sup>	0	0	0 <sup>*1</sup>	0 <sup>*1</sup>	0	0	0	0	0	0	0	0 <sup>*1</sup>	
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit field:	ASEL	ISEL	EOFR[1:0]	DSCR[1:0]	—	—	—	NCODR	—	PCR	—	PDR	PIDR	PODR			
Value after reset:	0	0	0	0	0	0 <sup>*1</sup>	0	0	0	0	0	0 <sup>*1</sup>	0	0	x	0	

Bit	Symbol	Function	R/W
0	PODR	Port Output Data 0: Low output 1: High output	R/W
1	PIDR	Pmn State 0: Low level 1: High level	R
2	PDR	Port Direction 0: Input (functions as an input pin) 1: Output (functions as an output pin)	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	PCR	Pull-up Control 0: Disable input pull-up 1: Enable input pull-up	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	NCODR	N-Channel Open-Drain Control 0: CMOS output 1: NMOS open-drain output	R/W
9:7	—	These bits are read as 0. The write value should be 0.	R/W
11:10	DSCR[1:0]	Port Drive Capability 0 0: Low drive 0 1: Middle drive 1 0: High-speed high-drive 1 1: High drive	R/W
13:12	EOFR[1:0]	Event on Falling/Event on Rising <sup>*2</sup> 0 0: Don't care 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
14	ISEL	IRQ Input Enable 0: Not used as an IRQn input pin 1: Used as an IRQn input pin	R/W
15	ASEL	Analog Input Enable 0: Not used as an analog pin 1: Used as an analog pin	R/W
16	PMR	Port Mode Control 0: Used as a general I/O pin 1: Used as an I/O port for peripheral functions	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
28:24	PSEL[4:0]	Peripheral Select These bits select the peripheral function. For individual pin functions, see the associated tables in this chapter.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note 1. The initial value of P201, P208, P209, P210, P211, P814 and P815 is not 0x0000\_0000. P201 is 0x0000\_0010, P208 is 0x0001\_0010, P209 is 0x0001\_0400, P210 is 0x0001\_0410, P211 is 0x0001\_0010, P814 is 0x1301\_0000 and P815 is 0x1301\_0000.

Note 2. Supported by PORTn (n = 1 to 4).

Port mn Pin Function Select Register (PmnPFS/PmnPFS\_HA/PmnPFS\_BY) is a 32-bit, 16-bit, or 8-bit read/write control register that selects the port mn pin function, and is accessed in 32-bit units. PmnPFS\_HA (PmnPFS [15:0] bits) is accessed in 16-bit units. PmnPFS\_BY (PmnPFS[7:0] bits) is accessed in 8-bit units.

The available Port mn pin depends on the product. For details, see [Table 19.1](#)

### PODR bit (Port Output Data), PIDR bit (Port State), PDR bit (Port Direction)

The PDR, PIDR, and PODR bits serve the same function as the PCNTR. When these bits are read, the PCNTR value is read.

### PCR bit (Pull-up Control)

The PCR bit enables or disables an input pull-up resistor on the individual port pins. When a pin is in the input state with the associated bit in PmnPFS.PCR set to 1, the pull-up resistor connected to the pin is enabled. When a pin is set as a general port output pin, or a peripheral function output pin, the pull-up resistor for the pin is disabled regardless of the PCR setting. The pull-up resistor is also disabled in the reset state. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

### NCODR bit (N-Channel Open-Drain Control)

The NCODR bit specifies the output type for the port pins. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

### DSCR[1:0] bits (Port Drive Capability)

The DSCR[1:0] bits switches the capability of the port. If the capability of a pin is fixed, the associated bit is a read/write bit, but the capability cannot be changed. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

### EOFR[1:0] bits (Event on Falling/Event on Rising)

The EOFR[1:0] bits select the edge detection method for the port group input signal. These bits support rising, falling, or both edge detections. When the EOFR[1:0] bits are set to 01b, 10b, or 11b, the input enable of the I/O cell is asserted. Following that, the event pulse is input from the external pin, and the GPIO outputs the event pulse to the ELC. Bits associated with non-existent pins are reserved. Reserved bits are read as 0. The write value should be 0.

### ISEL bit (IRQ Input Enable)

The ISEL bit specifies IRQ input pins. This setting can be used in combination with the peripheral functions, although an IRQn (external pin interrupt) of the same number must only be enabled for one pin. The ISEL bit for an unspecified IRQn is reserved.

### ASEL bit (Analog Input Enable)

The ASEL bit specifies analog pins. When a pin is set as an analog pin by this bit:

1. Specify it as a general I/O port in the Port Mode Control bit (PmnPFS.PMR).
2. Disable the pull-up resistor in the Pull-up Control bit (PmnPFS.PCR).
3. Specify the input in the Port Direction bit (PmnPFS.PDR). The pin state cannot be read at this point. The PmnPFS register is protected by the Write-Protect Register for Non-secure (PWPR\_NS). Release write-protect before modifying the register.

The ASEL bit for an unspecified analog I/O pin is reserved.

**PMR bit (Port Mode Control)**

The PMR bit specifies the port pin function. Bits associated with non-existent pins are reserved. The write value should be 0.

**PSEL[4:0] bits (Peripheral Select)**

The PSEL[4:0] bits assign the peripheral function.

**19.2.6 PWPR\_NS : Write-Protect Register for Non-secure**

Base address: PFS\_NS = 0x5040\_0800

Offset address: 0x50C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	PFSWE	PmnPFS Register Write Enable 0: Writing to the PmnPFS register is disabled 1: Writing to the PmnPFS register is enabled	R/W
7	B0WI	PFSWE Bit Write Disable 0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled	R/W

Note: S-TYPE-7, P-TYPE-2

Note: The access to this register is not controlled by any security attribution register.

**PFSWE bit (PmnPFS Register Write Enable)**

Writing to the PmnPFS register is enabled only when the PFSWE bit is set to 1. You must first write 0 to the B0WI bit before setting PFSWE to 1.

**B0WI bit (PFSWE Bit Write Disable)**

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

**19.2.7 PFENET : Ethernet Control Register**

Base address: PFS = 0x4040\_0800  
PFS\_NS = 0x5040\_0800

Offset address: 0x500

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PHYMODE0	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	PHYMODE0	Ethernet Mode Setting ch0 0: RMI mode (ETHERC channel 0) 1: MII mode (ETHERC channel 0)	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: The access to PHYMODE0 bit is controlled by PSARB.PSARB15 bit and PPARB.PPARB15 bit.

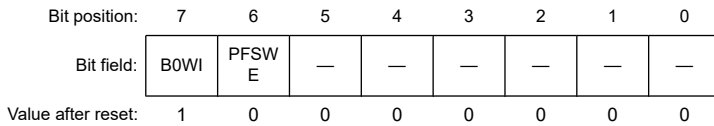
**PHYMODE0 bit (Ethernet Mode Setting ch0)**

The PHYMODE0 bit specifies the PHY mode of ETHERC channel 0. Select the same mode as that specified in the pin function select bits (PmnPFS.PSEL[4:0]). When the signals for the RMII mode are specified in the PmnPFS.PSEL[4:0] bits, set the PHYMODE bit to 0 (RMII mode). When the signals for the MII mode are specified in the PmnPFS.PSEL[4:0] bits, set the PHYMODE bit to 1 (MII mode).

**19.2.8 PWPR\_S : Write-Protect Register for Secure**

Base address: PFS = 0x4040\_0800

Offset address: 0x514



Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	PFSWE	PmnPFS Register Write Enable 0: Disable writes to the PmnPFS register 1: Enable writes to the PmnPFS register	R/W
7	B0WI	PFSWE Bit Write Disable 0: Enable writes the PFSWE bit 1: Disable writes to the PFSWE bit	R/W

Note: S-TYPE-6, P-TYPE-2

**PFSWE bit (PmnPFS Register Write Enable)**

Writing to the PmnPFS register of the IO port pin set as secure by the PmSAR register is enabled only when the PFSWE bit is set to 1. You must first write 0 to the B0WI bit before setting PFSWE to 1.

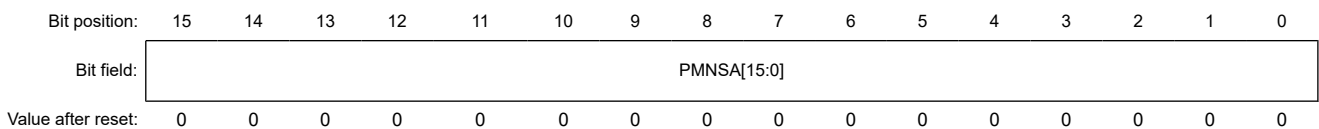
**B0WI bit (PFSWE Bit Write Disable)**

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

**19.2.9 PmSAR : Port Security Attribution register (m = 0 to 9)**

Base address: PFS = 0x4040\_0800  
PFS\_NS = 0x5040\_0800

Offset address: 0x530 + 0x004 × m



Bit	Symbol	Function	R/W
15:0	PMNSA[15:0]	Pmn Security Attribution Target I/O port pin : Pmn 0: Secure 1: Non Secure	R/W

Note: S-TYPE-1, P-TYPE-1

Note: This register is write-protected by PRCR register.

Note: m = 0 to 9, n = 00 to 15

Port Security Attribution Register is a 16-bit register that setting the Security Attribution of the each port, the registers are accessed only in 16-bit units.

**PMNSA[15:0] bits (Pmn Security Attribution)**

The PmnSA bit specifies the Security Attribution of Pmn.

**19.2.10 LVOCR : Low Voltage Operation Control register**

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xAB0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LVO1E	LVO0E

Value after reset: 0 0 0 0 0 0 0 1 1

Bit	Symbol	Function	R/W
0	LVO0E	Low Voltage Operation 0 Enable 0: Disable 1: Enable	R/W
1	LVO1E	Low Voltage Operation 1 Enable 0: Disable 1: Enable	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Low voltage operation control register controls the IO when VCC/VCC2 is lower than 2.7V.

**LVO0E bit (Low Voltage Operation 0 Enable)**

When VCC is lower than 2.7V, set the LVO0E bit to 1.

**LVO1E bit (Low Voltage Operation 1 Enable)**

When VCC2 is lower than 2.7V, set the LVO1E bit to 1.

**19.3 Operation****19.3.1 General I/O Ports**

All pins except P208 to P211, P814 and P815 operate as general I/O ports after reset. General I/O ports are organized as 16 bits per port and can be accessed by port with the Port Control Registers (PCNTRn, where n = 1 to 4), or by individual pins with the Port mn Pin Function Select register. For details on these registers, see [section 19.2. Register Descriptions](#).

Each port has the following bits:

- Port Security Attribution register (PmSAR)(m = 0 to 9), which indicates the security attribution.
- Port Direction bit (PDRn), which selects input or output direction
- Port Output Data bit (PODRn), which holds data for output
- Port Input Data bit (PIDRn), which indicates the pin states
- Event Input Data bit (EIDRn), which indicates the pin state when an ELC\_PORTn (n = 1, 2, 3 or 4) signal occurs
- Port Output Set bit (POSRn), which indicates the output value when a software write occurs
- Port Output Reset bit (PORRn), which indicates the output value when a software write occurs
- Event Output Set bit (EOSRn), which indicates the output value when an ELC\_PORTn (n = 1, 2, 3 or 4) signal occurs
- Event Output Reset bit (EORRn), which indicates the output value when an ELC\_PORTn (n = 1, 2, 3 or 4) signal occurs.

**19.3.2 Port Function Select**

The following port functions are available for configuring each pin:

- Security function: Security attribution for each pins
- I/O configuration: CMOS output or NMOS open-drain output, pull-up control, and drive capability
- General I/O port: Port direction, output data setting, and read input data
- Alternate function: Configured function mapping to the pin.

Each pin is associated with a Port mn Pin Function Select register (PmnPFS), which includes the associated PODR, PIDR, and PDR bits. In addition, the PmnPFS register includes the following:

- PCR: Pull-up resistor control bit that turns the input pull-up MOS on or off
- NCODR: N-channel open-drain control bit that selects the output type for each pin
- DSCR[1:0]: Drive capability control bit that selects the drive capability
- EOFR[1:0]: For selecting the edge of the event that input from the port group
- ISEL: IRQ input enable bit to specify an IRQ input pin
- ASEL: Analog input enable bit to specify an analog pin
- PMR: Port mode bit to specify the pin function of each port
- PSEL[4:0]: Port function select bits to select the associated peripheral function.

These configurations can be made by a single-register access to the Port mn Pin Function Select register. For details, see [section 19.2.5. PmnPFS/PmnPFS\\_HA/PmnPFS\\_BY : Port mn Pin Function Select Register \(m = 0 to 9, n = 00 to 15\)](#).

### 19.3.3 Port Group Function for ELC

In the MCU, Port 1 are assigned for the ELC port group function.

#### 19.3.3.1 Behavior When ELC\_PORTn (n = 1, 2, 3 or 4) is Input from ELC

The MCU supports the two functions described in this section when an ELC\_PORTn (n = 1, 2, 3 or 4) signal comes from the ELC.

##### (1) Input to EIDR

For the GPI function (PDR = 0 and PMR = 0 in the PmnPFS register), when an ELC\_PORTn (n = 1, 2, 3 or 4) signal comes from the ELC, the input enable of the I/O cell is asserted, and data from the external pins is read into the EIDR bit. See [Figure 19.2](#)

For the GPO function (PDR = 1) or the peripheral mode (PMR = 1), 0 is input into the EIDR bit from the external pins.

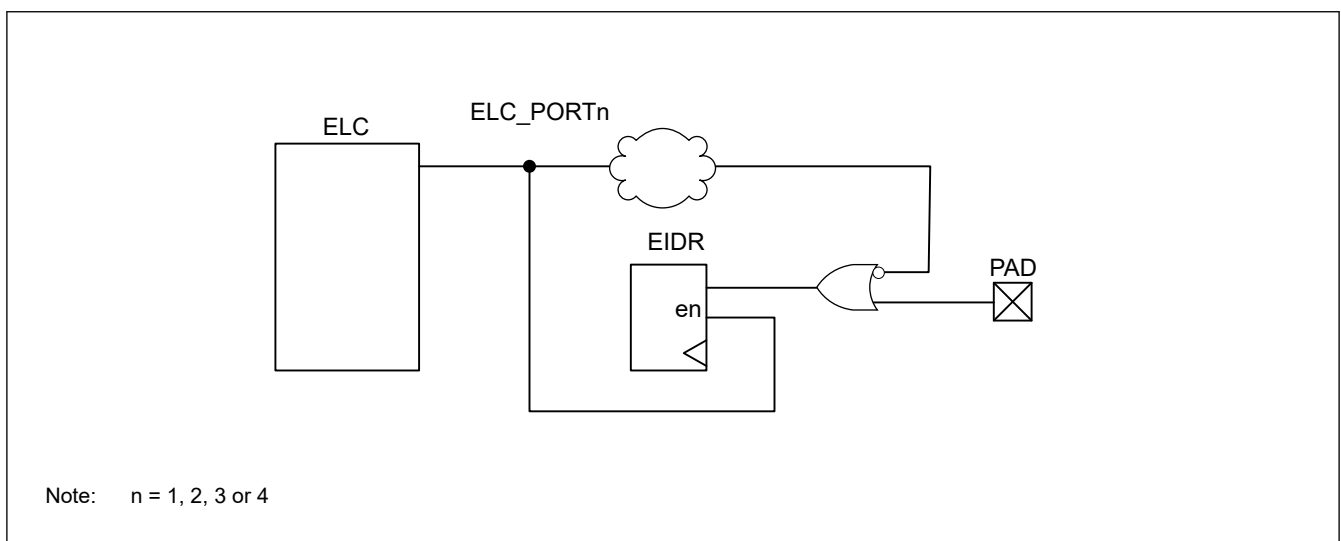
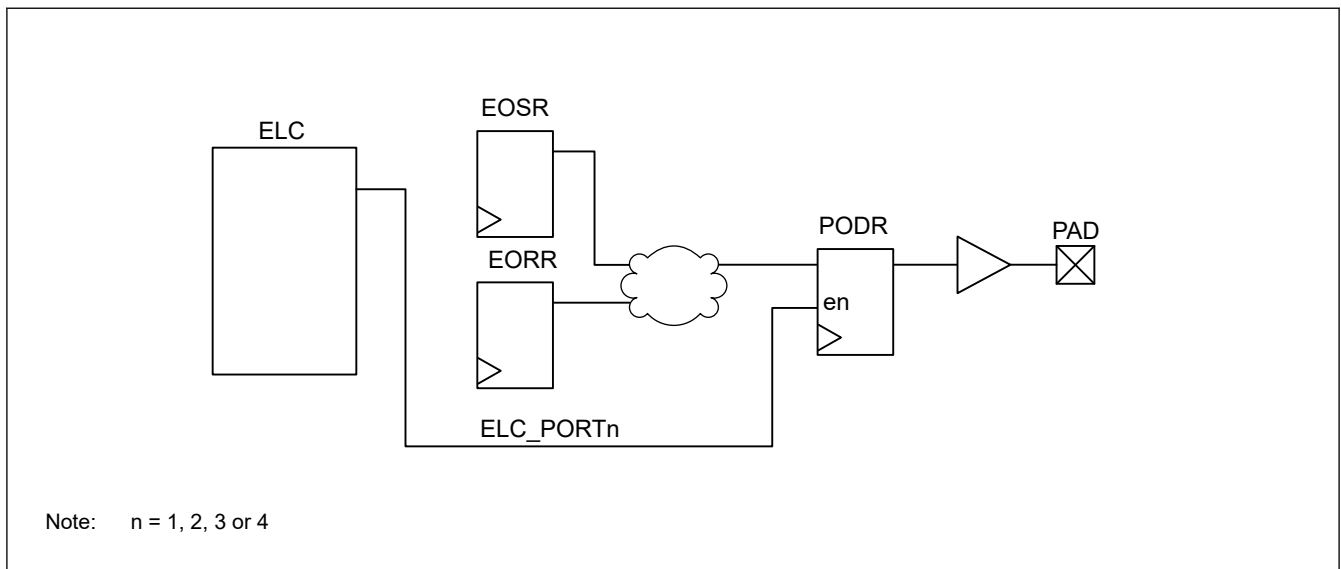


Figure 19.2 Event ports input data

## (2) Output from PODR by EOSR and EORR

When an ELC\_PORTn (n = 1, 2, 3 or 4) signal occurs, the data is output from the PODR to the external pin based on the settings in the EOSR and EORR registers.

- If EOSR is set to 1, when an ELC\_PORTn (n = 1, 2, 3 or 4) signal occurs, the PODR register outputs 1 to the external pin. Otherwise, when EOSR = 0, the PODR value is retained.
- If EORR is set to 1, when ELC\_PORTn (n = 1, 2, 3 or 4) signal occurs, the PODR register outputs 0 to the external pin. Otherwise, when EORR = 0, the PODR value is retained.



**Figure 19.3** Event ports output data

### 19.3.3.2 Behavior When an Event Pulse is Output to ELC

To output the event pulse from the external pins to the ELC, set the EOFR[1:0] bits in the PmnPFS register. For details, see [section 19.2.5. PmnPFS/PmnPFS\\_HA/PmnPFS\\_BY : Port mn Pin Function Select Register \(m = 0 to 9, n = 00 to 15\)](#). When the EOFR[1:0] bits are set, the input enable of the I/O cell is asserted.

Data from the external pin is the input. For example, for Port 1, when the data is input from P100 to P115, the data of those 16 pins is organized by OR logic. This data is formed into a one-shot pulse that goes to the ELC. The operation of Port n (n = 2 to 4) is also the same as Port 1. See [Figure 19.4](#).

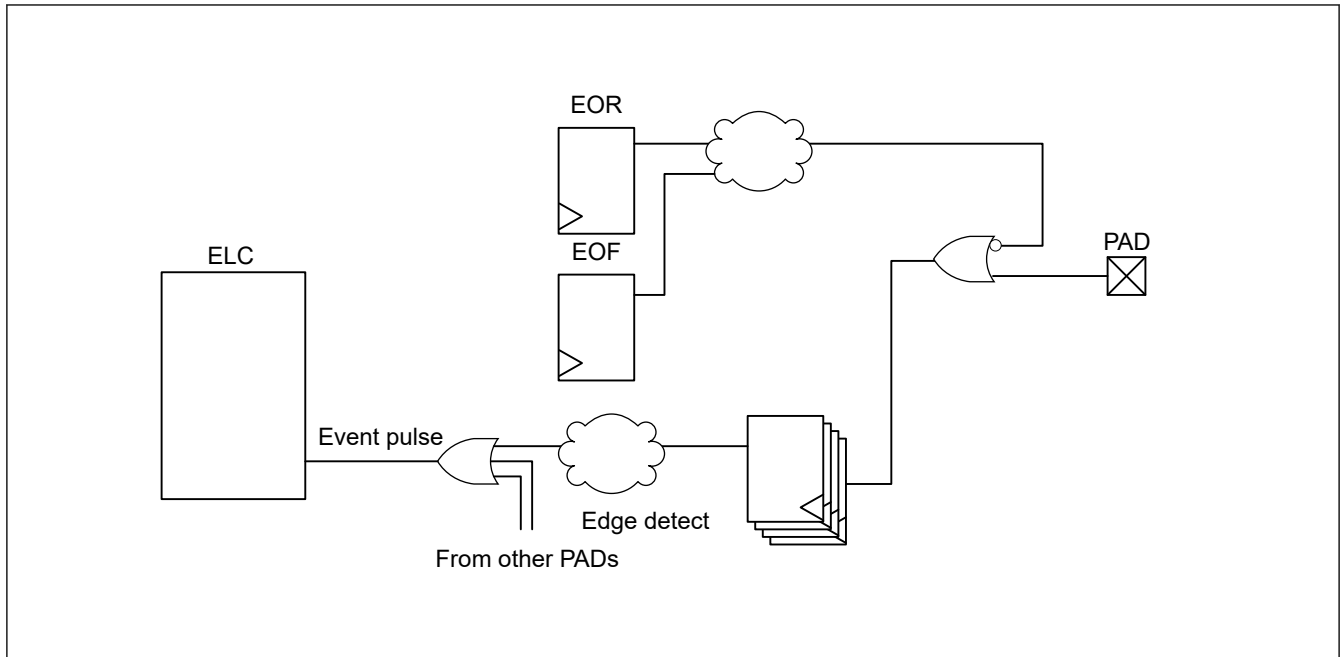


Figure 19.4 Generation of event pulse

## 19.4 Handling of Unused Pins

Table 19.3 shows how to handle unused pins.

Table 19.3 Handling of unused pins

Pin name	Description
MD	Use as a mode selection pin
RES	Connect to VCC through a resistor
USB_DP, USB_DM	When both P814PFS.PMR and P815PFS.PMR bits are set to 1, keep these pins open. When P814PFS.PMR or P815PFS.PMR bit is set to 0, configure it in the same way as Other ports.
P200/NMI	Connect to VCC through a resistor
EXTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P212). When this pin is not used as port P212, configure it in the same way as ports 1 to B.
XTAL	When the main clock oscillator is not used, set the MOSCCR.MOSTP bit to 1 (general port P213). When the external clock is input to the EXTAL pin, the XTAL pin functions as P213. When this pin is not used as port P213, configure it in the same way as ports 1 to B.
XCIN	Connect to VSS through a resistor
XCOU	Keep pin open
P000 to P010, P014, P015	<ul style="list-style-type: none"> <li>If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to AVCC0 through a resistor or to AVSS0 through a resistor *1</li> <li>If the direction is set to output (PCNTR1.PDRn = 1), keep pin open *1</li> </ul>
P100 to P107, P600, P800 to P804, P808, P809	<ul style="list-style-type: none"> <li>If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to VCC2 through a resistor or to VSS through a resistor *1</li> <li>If the direction is set to output (PCNTR1.PDRn = 1), keep pin open *1</li> </ul>
Other ports	<ul style="list-style-type: none"> <li>If the direction is set to input (PCNTR1.PDRn = 0), connect the associated pin to VCC through a resistor or to VSS through a resistor *1 *2</li> <li>If the direction is set to output (PCNTR1.PDRn = 1), keep pin open *1 *3</li> </ul>
VREFH0, VREFH	Connect to AVCC0
VREFL0, VREFL	Connect to AVSS0
VLO	Keep pin open *4
VBATT	Connect to VCC

Note 1. Clear the PmnPFS.PMR, PmnPFS.ISEL, PmnPFS.PCR, and PmnPFS.ASEL bits to 0.



Note 2. P208, P210 and P211 are recommended for pull up VCC through a resistor, because these pins are input pull-up enabled from the initial value (PmnPFS.PCR = 1).

Note 3. P209 is recommended for setting the direction to output (PCNTR1.PDRn = 1), because this pin is output from the initial value.

Note 4. For more information, see [section 51, Internal Voltage Regulator](#)

## 19.5 Usage Notes

### 19.5.1 Procedure for Specifying the Pin Functions

To specify the I/O pin functions:

1. Clear the B0WI bit in the PWPR\_NS register. This enables writing to the PFSWE bit in the PWPR\_NS register.\*<sup>1</sup>
2. Set 1 to the PFSWE bit in the PWPR\_NS register. This enables writing to the PmnPFS register.\*<sup>1</sup>
3. Clear the Port Mode Control bit in the PMR to 0 for the target pin to select the general I/O port.
4. Specify the I/O function for the pin through the PSEL[4:0] bits settings in the PmnPFS register.
5. Set the PMR bit to 1 as required to switch to the selected I/O function for the pin.
6. Clear the PFSWE bit in the PWPR\_NS register. This disables writing to the PmnPFS register.\*<sup>1</sup>
7. Set 1 to the B0WI bit in the PWPR\_NS register. This disables writing to the PFSWE bit in the PWPR\_NS register.\*<sup>1</sup>

Note 1. When the security attribution of Pmn is set to 0, set the PWPR\_S register to write to the PmnPFS register.

### 19.5.2 Procedure for Using Port Group Input

To use the port group input (port n (n = 1 to 4)):

1. Set the ELSRx.ELS[8:0] bits to all 0 to ignore unexpected pulses. For more information, see [section 18, Event Link Controller \(ELC\)](#).
2. Set the EOFR[1:0] bits of the PmnPFS register to specify the rising, falling, or both edge detections.
3. Execute a dummy read or wait for a short time, for example 100 ns. Ignoring of unexpected pulses depends on the initial value of the external pin.
4. Set the ELSRx.ELS[8:0] bits to enable the event signals.

### 19.5.3 Port Output Data Register (PODR) Summary

This register outputs data as follows:

1. Outputs 0 if PCNTR4.EORR is set to 1 when ELC\_PORTn (n = 1, 2, 3 or 4) signal occurs.
2. Outputs 1 if PCNTR4.EOSR is set to 1 when ELC\_PORTn (n = 1, 2, 3 or 4) signal occurs.
3. Outputs 0 if PCNTR3.PORR is set to 1.
4. Outputs 1 if PCNTR3.POSR is set to 1.
5. Outputs 0 or 1 because PCNTR1.PODRn is set.
6. Outputs 0 or 1 because PmnPFS.PODRn is set.

Numbers in this list correspond to the priority for writing to the PODRn. For example, if 1. and 3. from the list occur at the same time, the higher priority event 1. is executed.

### 19.5.4 Notes on Using Analog Functions

To use an analog function, set the Port Mode Control bit (PMR) and the Port Direction bit (PDRn) to 0 so that the pin acts as a general input port. Next, set the Analog Input Enable bit (ASEL) in the Port mn Pin Function Select Register (PmnPFS.ASEL) to 1.

### 19.5.5 I/O Buffer Specification

The P402, P403, and P404 can be used as the RTC input, tamper detection and other peripheral functions. [Table 19.4](#) lists the P402, P403, P404 specifications.

**Table 19.4 P402, P403, P404 specifications**

I/O port	RTC and tamper detection			Other peripheral	
	RTC and tamper detection input enable register	RTC	Tamper detection	Other peripheral enable register	CAC, CEU GPT, CAN, SCI, SSIE, ETHERC (MII), ETHERC (RMII), and interrupt
P402	VBTICTLR.VCH0INEN	RTCIC0	RTCIC0	P402PFS.PSEL and PMR	For details, see <a href="#">section 19.6. Peripheral Select Settings for Each Product</a> .
P403	VBTICTLR.VCH1INEN	RTCIC1	RTCIC1	P403PFS.PSEL and PMR	
P404	VBTICTLR.VCH2INEN	RTCIC2	RTCIC2	P404PFS.PSEL and PMR	

These RTC inputs and tamper detection are controlled by the VBTICTLR register and this register has the highest priority for selecting the functions.

P402, P403, and P404 can be used as IRQn-DS (n = 4, 14, 15) whether RTC inputs and tamper detection are selected or not. When using these interrupts, set the interrupt procedure after setting the VBTICTLR register. See [section 11.2.8. VBTICTLR : VBATT Input Control Register](#).

See [Figure 19.5](#).

The VBTICTLR register is only initialized by VBATT\_POR reset. Therefore, when not using the RTC inputs and tamper detection, the associated bit of VBTICTLR register must be set to 0 after except VBATT\_POR reset.

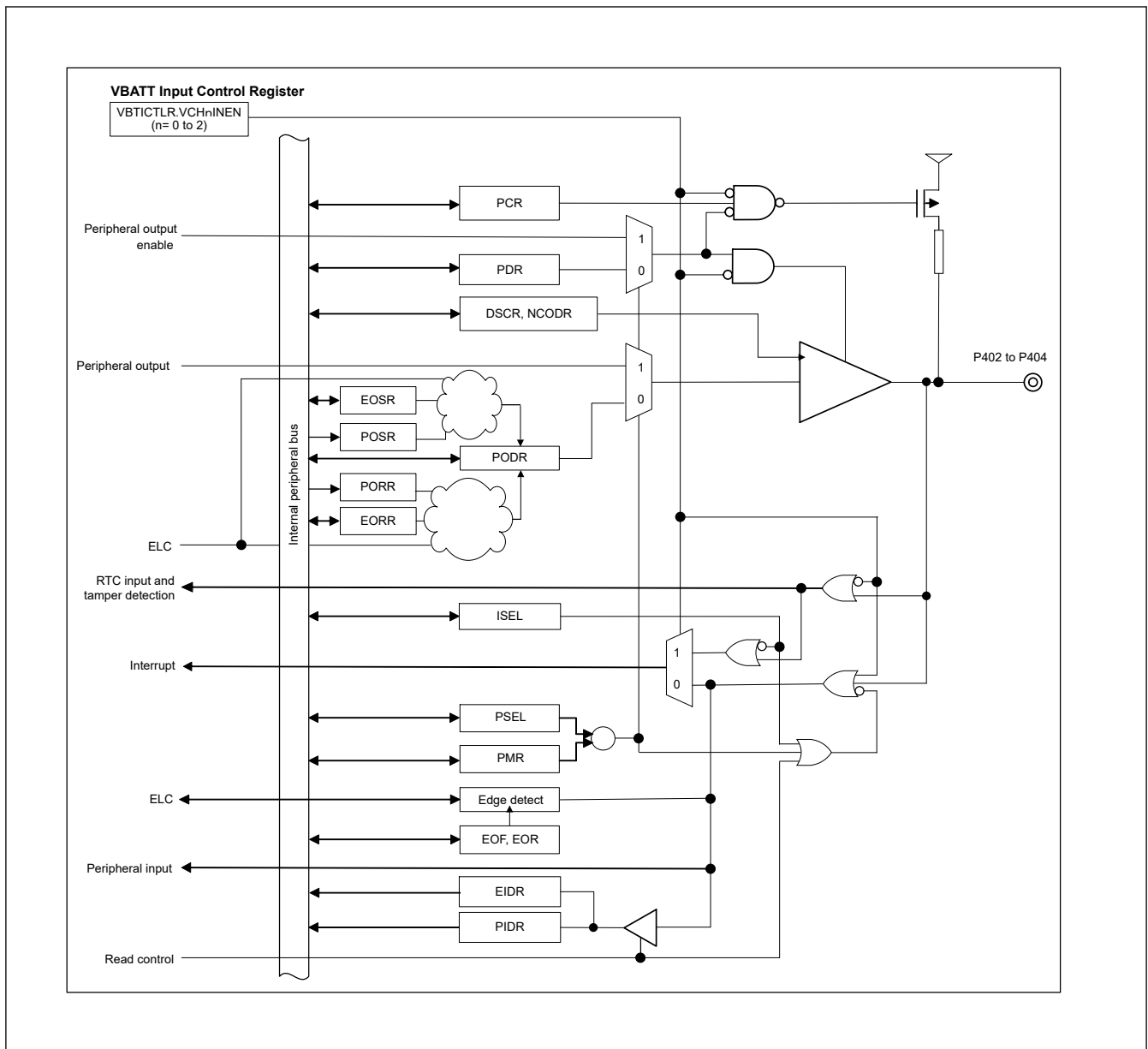


Figure 19.5 P402, P403, P404 diagram

### 19.5.6 Selecting Pins USB\_DP and USB\_DM

Pins USB\_DP and USB\_DM are shared with pins P814 and P815, respectively. Pins USB\_DP and P814 can be set with the PFS.P814PFS.PMR bit, and pins USB\_DM and P815 can be set with the PFS.P815PFS.PMR bit. Table 19.5 shows setting values of bits PFS.P814PFS.PMR and PFS.P815PFS.PMR with each selected pin.

Table 19.5 Selecting the USB/PORT pins

PMR Bits Settings		Pins Selected	
P814PFS.PMR bit	P815PFS.PMR bit	P814/USB_DP Pin	P815/USB_DM Pin
0	0	P814	P815
0	1	P814	P815
1	0	P814	P815
1	1	USB_DP	USB_DM

Note: When using P814/USB\_DP and P815/USB\_DM as GPIO pins (P814 and P815), use the USB related registers with their initial values.

- Note: When using P814/USB\_DP and P815/USB\_DM as USB pins (USB\_DP and USB\_DM), use the GPIO related registers for P814 and P815 with their initial values.
- Note: When using P814/USB\_DP and P815/USB\_DM as GPIO pins or USB pins, set these pins only once after a reset.

### 19.5.7 Pull-up/Pull-down Setting for P814 and P815 at USBFS/GPIO Function

When the GPIO function of P814 and P815 is used, the Pull-up/Pull-down function of USBFS register affects to P814 and P815. Therefore, before using GPIO function, the pull-up and pull-down control of USBFS needs to be disabled. The control register is SYSCFG.DPRPU and SYSCFG.DRPD bits.

## 19.6 Peripheral Select Settings for Each Product

This section describes the pin function select configuration using the PmnPFS register. Some pin names have added \_A, \_B, or \_C suffixes. When assigning IIC, SPI, SSIE and ETHERC functionality, select the functional pins having the same suffix. The other pins can be selected regardless of the suffix. Assigning the same function to two or more pins simultaneously is prohibited.

- In Pmn pin function select register(PmnPFS), the PSEL bits have to be set when the PMR bit of the target pin is 0. If the PSEL bits are set when the PMR bit is 1, the unexpected edges may be input at the input function or the unexpected pulses may be output to the external pin at the output function.
- Only the allowed values (functions) should be specified in the PSEL bits of PmnPFS register. If a value which is not allowed for the register is specified, the correct operation is not guaranteed.
- The single function should not be assigned to the multiple pins by PmnPFS register. When the GPT1, GPT5, SCI3, IIC0 or SPI0 are configured as secure and these pin function is being assigned to the pin which security attribution is set as secure by the PmSAR register, the write access to the PSEL bits for setting same function as secure pin in other pins is ignored when the security attribution of that pin is non-secure. For example, if the PSARE.PSARE30 bit is 0 (GPT1 is secure) and the P209PFS.PSEL bits is 00011b (pin function is GTIOC1A) and the P2SAR.109SA bit is 0 (P209 is secure), the write 00011b to the P405PFS.PSEL bits is ignored when the P4SAR.405SA bit is 1 (P405 is non-secure).
- The PORT0, PORT5 and PORT8 have the analog functions such as A/D converter. When these pins are used as an analog function, for avoiding the loss of resolution, the PMR bit should be set to 0 and PDR bit should be set to 0. After that, ASEL bit should be set to 1

**Table 19.6 Register settings for input/output pin function (PORT0)**

PSEL[4:0] settings	Function	pin													
		P000	P001	P002	P003	P004	P005	P006	P007	P008	P009	P010	P014	P015	
00000b (value after reset)	Hi-Z/ JTAG/SWD	Hi-Z													
ASEL bit		AN100/ IVCMP2	AN101/ IVREF0	AN102/ IVCMP3	AN104/ IVREF1	AN000/ IVCMP2	AN001	AN002/ IVCMP3	AN004	AN008	AN006	AN005/ IVCMP0	AN007/ DA0	AN105	
ISEL bit		IRQ6-DS	IRQ7-DS	IRQ8-DS	—	IRQ9-DS	IRQ10- DS	IRQ11- DS	—	IRQ12- DS	IRQ13- DS	IRQ14	—	IRQ13	
DSCR[1:0] bits	Drive capability contro <sup>1</sup>	L	L	L	L	L	L	L	L	L	L	L	L	L	
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
144 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
100 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	✓	✓	

✓: Available  
 —: Setting prohibited

Note 1. The drive strength of this port cannot be controlled by PmnPFS.DSCR[1:0] bits.

**Table 19.7 Register settings for input/output pin function (PORT1) (1 of 2)**

PSEL[4:0] settings	Function	pin													
		P100	P101	P102	P103	P104	P105	P106	P107	P112	P113	P114	P115		
00000b(value after reset)	Hi-Z/ JTAG/SW D	Hi-Z													
00001b	AGT	AGTIO0	AGTEE0	AGTO0	—	—	—	AGTOB0	AGTOA0	—	—	—	—	—	
00010b	GPT <sup>2</sup>	GTETRGA	GTETRGB	—	—	GTETRGB	—	—	—	—	—	—	—	—	

**Table 19.7 Register settings for input/output pin function (PORT1) (2 of 2)**

PSEL[4:0] settings	Function	pin												
		P100	P101	P102	P103	P104	P105	P106	P107	P112	P113	P114	P115	
00011b	GPT <sup>2</sup>	—	—	GTIOC2B	GTIOC2A	GTIOC1B	GTIOC1A	—	—	GTIOC3B	GTIOC2A	GTIOC2B	GTIOC5A	
00100b	SCI <sup>1</sup>	—	—	—	—	—	—	—	—	TXD0_A/ MOSI0_A/ SDA0_A	RXD0_A/ MISO0_A/ SCL0_A	CTS_RT S0_A/ SS0_A	CTS0_A	
00101b	SCI <sup>1</sup>	SCK9_A	RXD9_A/ MISO9_A/ SCL9_A	TXD9_A/ MOSI9_A/ SDA9_A	CTS_RT S9_A/ SS9_A	CTS9_A	—	—	—	—	—	—	—	
00110b	SPI <sup>1</sup>	MISOB_A	MOSIB_A	RSPCKB_A	SSLB0_A	SSLB1_A	SSLB2_A	SSLB3_A	—	SSLA2_B	SSLA1_B	SSLA0_B	MOSIA_B	
01010b	CAC/ ADC12	—	—	ADTRG 0	—	—	—	—	—	—	—	—	—	
01101b	SCI	—	—	—	—	—	—	—	—	—	—	DE0	—	
01110b	SCI	DE9	—	—	DE9	—	—	—	—	—	—	—	—	
10000b	CANFD	—	—	CRX0	CTX0	—	—	—	—	—	—	—	—	
10010b	SSIE <sup>1</sup>	—	—	—	—	—	—	—	—	SSIBCK0_B	SSILRC K0/ SSI FS0_B	SSIRX D0_B	SSITX D0_B	
10110b	ETHERC (MII)	—	—	—	—	—	—	—	—	ET0_CRS	ET0_EXO UT	ET0_LINK STA	ET0_WOL	
10111b	ETHERC (RMII)	—	—	—	—	—	—	—	—	RMII0_CR S_DV_A	ET0_EXO UT	ET0_LINK STA	ET0_WOL	
11100b	OSPI	OM_SIO0	OM_SIO3	OM_SIO4	OM_SIO2	OM_CS1	OM_EC SINT1	OM_RESE T	OM_CS0	—	—	—	—	
11110b	ULPT	—	—	—	—	—	ULPT01- DS	ULPTEE1- DS	—	ULPT0B0- DS	ULPT0A0- DS	—	—	
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	
ISEL bit		IRQ2	IRQ1	—	—	IRQ1	IRQ0	—	—	—	—	—	—	
DSCR[1:0] bits	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
144 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
100 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

✓: Available  
—: Setting prohibited

- Note 1. Recommend using pins that have a letter appended to their names, for instance “\_A”, “\_B”, or “\_C” to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.
- Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t<sub>GTISK</sub>).

**Table 19.8 Register settings for input/output pin function (PORT2) (1 of 2)**

PSEL[4:0] settings	Function	Pin												
		P200 <sup>4</sup>	P201	P202	P203	P204	P205	P206	P208	P209	P210	P211	P212	P213
00000b (value after reset)	Hi-Z/ JTAG/SWD	Hi-Z							TDI	TDO	TMS/ SWDIO	TCK/ SWCLK	Hi-z	
00001b	AGT	—	—	—	—	AGTIO1	AGTO1	—	—	—	—	—	AGTEE 1	—
00010b	GPT <sup>2</sup>	—	—	—	—	—	—	—	—	—	—	—	GTETRG D	GTETRG C
00011b	GPT <sup>2</sup>	—	—	GTIOC5 B	GTIOC5 A	GTIOC4 B	GTIOC4 A	—	GTIOC1B	GTIOC1A	GTIOC0B	GTIOC0A	GTIOC0 B	GTIOC0 A
00100b	SCI <sup>1</sup>	—	—	—	—	SCK4_A	TXD4_A/ MOSI4_A / SDA4_A	RXD4_A/ MISO4_A / SCL4_A	—	—	—	—	—	—
00101b	SCI <sup>1</sup>	—	—	—	—	—	—	—	RXD9_B/ MISO9_B / SCL9_B	TXD9_B/ MOSI9_B / SDA9_B	CTS_RT S9_B/ SS9_B	SCK9_B	RXD1_C/ MISO1_C / SCL1_C	TXD1_C/ MOSI1_C / SDA1_C
00110b	SPI <sup>1</sup>	—	—	MOSIA_A	RSPCKA_A	SSLA0_A	SSLA1_A	SSLA2_A	—	—	—	—	—	—
00111b	IIC <sup>1</sup>	—	—	—	—	—	SCL1_B	SDA1_B	—	—	—	—	—	—

**Table 19.8 Register settings for input/output pin function (PORT2) (2 of 2)**

PSEL[4:0] settings	Function	Pin													
		P200*4	P201	P202	P203	P204	P205	P206	P208	P209	P210	P211	P212	P213	
01001b	CLKOUT/ACMPHS/RTC	—	—	—	—	—	—	CLKOUT	—	VCOUT	CLKOUT	—	—	—	—
01010b	CAC/ADC12	—	—	—	—	—	CACREF	—	—	—	—	—	—	ADTRG1	
01101b	SCI	—	—	—	—	—	DE4	—	—	—	—	—	—	—	
01110b	SCI	—	—	—	—	—	—	—	—	—	—	DE9	DE9	—	
10000b	CANFD	—	—	CRX0	CTX0	—	—	—	—	CRX1	CTX1	—	—	—	
10010b	SSIE*1	—	—	—	—	—	SSIBCK1_A	SSILRCK1/SSIFS1_A	SSIDATA1_A	—	—	—	—	—	
10011b	USBFS	—	—	—	—	—	USB_OVRCURB	USB_OVRCURA	USB_VBUSEN	—	—	—	—	—	
11110b	ULPT	—	—	ULPTO B1	ULPTO A1	—	—	—	—	—	—	—	—	ULPTEE0	
Don't-care	Trace(Debug)/Clock	—	—	—	—	—	—	—	—	—	SWO	—	—	EXTAL XTAL	
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	
ISEL bit		—	—	IRQ3-DS	IRQ2-DS	—	IRQ1-DS	IRQ0-DS	IRQ3	—	—	—	IRQ3	IRQ2	
DSCR[1:0] bits	Drive capability control	—	L*3	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	
NCODR bit	N-ch open-drain	—	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
144 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
100 pins product		✓	✓	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	

✓: Available  
 —: Setting prohibited

- Note 1. Recommend using pins that have a letter appended to their names, for instance “\_A”, “\_B”, or “\_C”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.
- Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t<sub>GTISK</sub>).
- Note 3. The driver strength of this port can not be controlled by PmnPFS.DSCR[1:0] bits.
- Note 4. When using NMI pin interrupt, Port related registers setting are not required.

**Table 19.9 Register settings for input/output pin function (PORT3) (1 of 2)**

PSEL[4:0] settings	Function	Pin														
		P300	P301	P302	P303	P304	P305	P306	P307	P308	P309	P310	P311	P312	P313	
00000b(value after reset)	Hi-Z/JTAG/SWD	Hi-z														
00001b	AGT	—	AGTIO0	—	—	—	—	—	—	—	—	—	AGTEE1	AGTOB1	AGTOA1	—
00010b	GPT*2	—	—	—	—	—	—	—	—	—	—	—	—	GTADM1	GTADM0	—
00011b	GPT*2	GTIOC3A	GTIOC4B	GTIOC4A	—	—	—	—	—	—	—	—	—	—	—	—
00100b	SCI*1	SCK0_A	—	—	—	—	—	—	—	—	—	—	—	—	—	—
00101b	SCI*1	—	—	—	—	—	—	—	—	—	CTS9_B	RXD3_B/MISO3_B/SCL3_B	TXD3_B/MOSI3_B/SDA3_B	SCK3_B	CTS_RTS3_B/SS3_B	CTS3_C
00110b	SPI*1	SSLA3_B	—	—	—	—	—	—	—	—	—	—	—	—	—	MISO_A
01010b	CAC/ADC12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
01101b	SCI	DE0	—	—	—	—	—	—	—	—	—	—	—	—	—	—
01110b	SCI	—	—	—	—	—	—	—	—	—	—	—	—	DE3	DE3	—
10000b	CANFD	—	—	—	—	—	—	—	—	—	—	—	—	CRX0	CTX0	—
10110b	ETHERC(MII)	ET0_RX_CLK	ET0_ERXD0	ET0_ERXD1	ET0_ETXD0	ET0_ETXD1	ET0_RX_ER	ET0_TX_EN	ET0_MDIO	ET0_MDIO	ET0_ETXD3	ET0_ETXD2	ET0_ETXD1	ET0_ETXD0	ET0_TX_CLK	—

**Table 19.9 Register settings for input/output pin function (PORT3) (2 of 2)**

PSEL[4:0] settings	Function	Pin													
		P300	P301	P302	P303	P304	P305	P306	P307	P308	P309	P310	P311	P312	P313
10111b	ETHERC (RMII)	RMII0_RX_ER_A	RMII0_RXD1_A	RMII0_RXD0_A	REF50CK0_A	RMII0_TXD0_A	RMII0_TXD1_A	RMII0_TXD_EN_A	ET0_MDIO	ET0_MD_C	—	—	—	—	—
11110b	ULPT	ULPTEVI0-DS	ULPTEE0-DS	ULPTO0-DS	—	ULPTO1	ULPTEE1	ULPTEVI1	ULPTOA1	ULPTOB1	—	—	—	—	—
Don't-care	Trace(Debug)	—	—	—	—	TDATA3	TDATA2	TDATA1	TDATA0	TCLK	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ4	IRQ6	IRQ5	—	IRQ9	IRQ8	—	—	—	—	—	—	—	—
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H/H	L/M/H/H	L/M/H/H	L/M/H/H	L/M/H/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
144 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
100 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	—

✓: Available  
 —: Setting prohibited

- Note 1. Recommend using pins that have a letter appended to their names, for instance “\_A”, “\_B”, or “\_C”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.
- Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t<sub>GTISK</sub>).

**Table 19.10 Register settings for input/output pin function (PORT4) (1 of 2)**

PSEL[4:0] settings	Function	pin																
		P400	P401	P402	P403	P404	P405	P406	P407	P408	P409	P410	P411	P412	P413	P414	P415	
00000b (value after reset)	Hi-Z/JTAG/SWD	Hi-Z																
00001b	AGT	AGTIO1	—	—	—	—	AGTIO1	—	AGTIO0	—	—	AGTOB1	AGTOA1	AGTEE1	—	—	—	
00010b	GPT <sup>3</sup>	—	GTETPGA	—	—	—	—	—	—	—	—	—	—	—	—	GTADM1	GTADM0	
00011b	GPT <sup>3</sup>	—	—	—	GTIOC3A	GTIOC3B	GTIOC1A	GTIOC1B	GTIOC10B	GTIOC10A	—	—	—	—	—	GTIOC0B	GTIOC0A	
00100b	SCI <sup>2</sup>	—	—	—	—	—	SCK2_B	TXD2_B/ MOSI2_B/ SDA2_B	CTS_RTS4_A/ SS4_A	CTS4_A	—	—	—	—	—	RXD4_B/ MISO4_B/ SCL4_B	TXD4_B/ MOSI4_B/ SDA4_B	
00101b	SCI <sup>2</sup>	TXD1_A/ MOSI1_A/ SDA1_A	RXD1_A/ MISO1_A/ SCL1_A	SCK1_A	CTS_RTS4_A/ SS4_A	CTS1_A	—	—	—	RXD3_A/ MISO3_A/ SCL3_A	TXD3_A/ MOSI3_A/ SDA3_A	SCK3_A	CTS_RTS3_A/ SS3_A	CTS3_A	—	—	—	
00110b	SPI <sup>2</sup>	—	—	—	—	—	—	—	SSLA3_C	SSLA3_A	—	—	MISOB_B	MOSIB_B	RSPCK_B	SSLB0_B	SSLB1_B	SSLB2_B
00111b	IIC <sup>2</sup>	—	—	—	—	—	—	—	SDA0_B	SCL0_B	SDA0_A	SCL0_A	—	—	—	—	—	—
01001b	CLKOUT/ACMPHS/RTC	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
01010b	CAC/ADC12	ADTRG1	—	CACREF	—	—	—	—	—	ADTRG0	—	—	—	—	—	—	—	—
01101b	SCI	—	—	—	—	—	—	DE2	—	DE4	—	—	—	—	—	—	—	—
01110b	SCI	—	—	DE1	DE1	—	—	—	—	—	—	—	DE3	DE3	—	—	—	—
01111b	CEU	VIO_D0	VIO_D1	—	—	VIO_D3	VIO_D2	VIO_D3	—	—	—	—	—	—	—	—	—	—
10000b	CANFD	—	CTX0	CRX0	—	—	—	—	—	—	—	—	—	—	—	—	CRX1	CTX1
10010b	SSIE <sup>2</sup>	AUDIO_CLK	—	AUDIO_CLK	SSIBCK0_A	SSILRC0/ SSIFS0_A	SSITXD0_A	SSIRXD0_A	—	—	—	—	—	—	—	—	—	—

**Table 19.10 Register settings for input/output pin function (PORT4) (2 of 2)**

PSEL[4:0] settings	Function	pin																
		P400	P401	P402	P403	P404	P405	P406	P407	P408	P409	P410	P411	P412	P413	P414	P415	
10011b	USBFS	—	—	—	—	—	—	—	—	USB_VBUS <sup>-</sup>	USB_VBUSEN	USB_OVRCUR A-DS	USB_OVRCUR B-DS	USB_ID	USB_EXICEN	—	—	—
10110b	ETHERC (MII)	ET0_WOL	ET0_MDC	ET0_MDIO	ET0_LI NKSTA	ET0_EXOUT	ET0_TXEN	ET0_RXER	—	—	—	—	—	—	—	—	—	—
10111b	ETHERC (RMII)	ET0_WOL	ET0_MDC	ET0_MDIO	ET0_LI NKSTA	ET0_EXOUT	RMII0_TXD_EN_B	RMII0_TXD1_B	—	—	—	—	—	—	—	—	—	—
11110b	ULPT	—	—	—	—	—	—	—	—	ULPTO B0	ULPTO A0	—	—	—	—	ULPTE E1	—	—
Don't-care	RTC/Clock	—	—	RTCIC0 <sup>*1</sup>	RTCIC1 <sup>*1</sup>	RTCIC2 <sup>*1</sup>	—	EXCIN	—	—	—	—	—	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ0	IRQ5-DS	IRQ4-DS	IRQ14-DS	IRQ15-DS	—	—	—	IRQ7	IRQ6	IRQ5	IRQ4	—	—	IRQ9	IRQ8	—
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
144 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
100 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

✓: Available  
 —: Setting prohibited

- Note 1. To use this pin function, set the associated pin as a general input (set the PmnPFS.PDR and PmnPFS.PMR bits to 0).  
 Note 2. Recommend using pins that have a letter appended to their names, for instance “\_A”, “\_B”, or “\_C”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.  
 Note 3. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t<sub>GTISK</sub>).

**Table 19.11 Register settings for input/output pin function (PORT5)**

PSEL[4:0] settings	Function	pin		
		P511	P512	P513
00000b (value after reset)	Hi-Z/ JTAG/SWD	Hi-Z	—	—
00011b	GPT <sup>2</sup>	GTIOC0B	GTIOC0A	—
00111b	IIC	SDA1_A	SCL1_A	—
01010b	CAC/ADC12	—	—	—
01111b	CEU	—	—	VIO_FLD
10000b	CANFD	CRX1	CTX1	—
10011b	USBFS	—	—	—
11110b	ULPT	—	—	—
ASEL bit		—	—	IVCMP0
ISEL bit		IRQ15	IRQ14	—
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓
144 pins product		✓	✓	✓
100 pins product		—	—	—

✓: Available  
 —: Setting prohibited

- Note 1. Recommend using pins that have a letter appended to their names, for instance “\_A”, “\_B”, or “\_C”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.  
 Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t<sub>GTISK</sub>).



**Table 19.12 Register settings for input/output pin function (PORT6)**

PSEL[4:0]settings	Function	Pin											
		P600	P601	P602	P603	P604	P605	P609	P610	P611	P612	P613	P614
00000b(value after reset)	Hi-Z/JTAG/SW D	Hi-Z											
00001b	AGT	—	—	—	—	—	—	—	—	—	—	AGT01	AGT00
00010b	GPT <sup>*2</sup>	—	—	—	—	—	—	—	—	—	—	GTETRGA	GTETRGB
00011b	GPT <sup>*2</sup>	—	—	—	—	—	—	—	GTIOC5B	GTIOC4A	GTIOC4B	—	—
00100b	SCI <sup>*1</sup>	—	SCK0_B	RXD0_B/ MISO0_B/ SCL0_B	TXD0_B/ MOSI0_B/ SDA0_B	CTS_RTS 0_B/ SS0_B	CTS0_B	TXD0_C/ MOSI0_C/ SDA0_C	RXD0_C/ MISO0_C/ SCL0_C	SCK0_C	CTS_RTS 0_C/ SS0_C	CTS0_C	—
00110b	SPI <sup>*1</sup>	—	—	—	—	—	—	MISOA_B	RSPCKA_B	MOSIA_B	SSLA0_B	—	—
01001b	CLKOUT/ ACMPHS/RTC	—	RTCOUT	—	—	—	—	—	—	CLKOUT	—	—	—
01010b	CAC/ADC12	CACREF	—	—	—	—	—	—	—	CACREF	—	—	—
01101b	SCI	—	DE0	—	—	DE0	—	—	—	DE0	DE0	—	—
10000b	CANFD	—	—	—	—	—	—	CTX1	CRX1	—	—	—	—
10011b	USBFS	—	—	—	—	—	—	—	—	—	—	—	—
10110b	ETHERC (MII)	—	—	—	—	—	—	ET0_RX_DV	ET0_COL	ET0_ERX_D2	ET0_ERX_D3	—	—
11100b	OSPI	OM_RST01	OM_WP1	—	—	—	—	—	—	—	—	—	—
11110b	ULPT	ULPTEV11-DS	ULPTEV10	ULPTEE0	ULPT00	—	—	ULPTOA1-DS	ULPTOB1-DS	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		—	—	—	—	—	—	—	—	—	—	—	—
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
144 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
100 pins product		✓	—	—	—	—	—	✓	✓	—	—	—	—

✓: Available  
 —: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance “\_A”, “\_B”, or “\_C”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (tGTISK).

**Table 19.13 Register settings for input/output pin function (PORT7) (1 of 2)**

PSEL[4:0]settings	Function	Pin											
		P700	P701	P702	P703	P704	P705	P708	P709	P710	P711	P712	P713
00000b(value after reset)	Hi-Z/JTAG/SW D	Hi-Z											
00001b	AGT	—	—	—	AGT01	AGT00	AGTIO0	—	—	—	AGTEE0	AGTOB0	AGTOA0
00010b	GPT <sup>*2</sup>	—	—	—	—	GTADSM0	GTADSM1	—	—	—	—	—	—
00011b	GPT <sup>*2</sup>	GTIOC5A	GTIOC5B	—	—	—	—	—	—	—	—	GTIOC2B	GTIOC2A
00100b	SCI <sup>*1</sup>	RXD2_B/ MISO2_B/ SCL2_B	CTS_RTS 2_B/ SS2_B	CTS2_B	—	—	—	SCK4_B	CTS_RTS 4_B/ SS4_B	CTS4_B	—	—	—
00101b	SCI <sup>*1</sup>	—	—	—	—	—	CTS1_B	—	—	—	—	—	—
00110b	SPI <sup>*1</sup>	MISOA_C	MOSIA_C	RSPCKA_C	SSLA0_C	SSLA1_C	SSLA2_C	SSLB3_B	—	—	—	—	—
01001b	CLKOUT/ ACMPHS/RTC	—	—	—	VCOU	—	—	—	—	—	—	—	—
01010b	CAC/ADC12	—	—	—	—	—	—	CACREF	—	—	—	—	—
01101b	SCI	—	DE2	—	—	—	—	DE4	DE4	—	—	—	—
01111b	CEU	VIO_D4	VIO_D5	VIO_D6	VIO_D7	—	—	VIO_CLK	VIO_HD	VIO_VD	—	—	—
10000b	CANFD	—	—	—	—	CTX0	CRX0	—	—	—	—	—	—

**Table 19.13 Register settings for input/output pin function (PORT7) (2 of 2)**

PSEL[4:0]settings	Function	Pin											
		P700	P701	P702	P703	P704	P705	P708	P709	P710	P711	P712	P713
10010b	SSIE <sup>1</sup>	SSIDATA1_B	SSILRCK1 / SSIFS1_A	SSIBCK1_B	—	—	—	AUDIO_CLK	—	—	—	—	—
10110b	ETHERC (MII)	ET0_ETXD1	ET0_ETXD0	ET0_ERXD1	ET0_ERXD0	ET0_RX_CLK	ET0_CRSD	—	—	—	—	—	—
10111b	ETHERC (RMII)	RMII0_TXD0_B	REF50CK0_B	RMII0_RXD0_B	RMII0_RXD1_B	RMII0_RX_ER_B	RMII0_CRSDV_B	—	—	—	—	—	—
11110b	ULPT	—	ULPTO1	ULPTO0	—	—	—	—	—	—	—	—	—
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		—	—	—	—	—	—	IRQ11	IRQ10	—	—	—	—
DSCR[1:0] bits	Drive capability control	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
144 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
100 pins product		—	—	—	—	—	—	✓	—	—	—	—	—

✓: Available  
 —: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance “\_A”, “\_B”, or “\_C”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t<sub>GTISK</sub>).

**Table 19.14 Register settings for input/output pin function (PORT8)**

PSEL[4:0]settings	Function	Pin											
		P800	P801	P802	P803	P804	P805	P806	P808	P809	P814	P815	
00000b(value after reset)	Hi-Z/JTAG/SWD	Hi-Z											
00001b	AGT	AGTOA0	AGTOB0	—	—	—	—	—	—	—	—	—	—
00010b	GPT <sup>2</sup>	—	—	—	GTETRGC	GTETRGD	—	—	—	—	—	—	—
00011b	GPT <sup>2</sup>	GTIOC11A	GTIOC11B	GTIOC12A	GTIOC12B	GTIOC13A	—	—	GTIOC13B	—	—	—	—
00100b	SCI <sup>1</sup>	CTS2_A	TXD2_A / MOSI2_A / SDA2_A	RXD2_A / MISO2_A / SCL2_A	SCK2_A	CTS_RTS2_A / SS2_A	—	—	—	—	—	—	—
01101b	SCI	—	—	—	DE2	DE2	—	—	—	—	—	—	—
01111b	CEU	—	—	—	—	—	—	—	—	—	—	—	—
10000b	CANFD	—	—	—	—	—	—	—	—	—	—	CRX0	CTX0
10011b	USBFS	—	—	—	—	—	—	—	—	—	—	—	—
11100b	OSPI	OM_SIO5	OM_DQS	OM_SIO6	OM_SIO1	OM_SIO7	—	—	OM_SCLK	OM_SCLKN	—	—	—
11110b	ULPT	—	—	—	—	—	—	—	—	—	—	—	—
Don't-care		USBFS	—	—	—	—	—	—	—	—	—	UBS_DP	UBS_DM
ASEL bit		—	—	—	—	—	—	—	—	—	—	—	—
ISEL bit		IRQ11	IRQ12	—	—	IRQ14	—	IRQ0	IRQ15	—	—	—	—
DSCR[1:0] bits	Drive capability control	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H/HH	L/M/H	L/M/H	L/M/H/HH	L/M/H/HH	L/M/H	L/M/H	L/M/H
NCODR bit	N-ch open-drain	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
PCR bit	Pull-up	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
144 pins product		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
100 pins product		✓	✓	✓	✓	✓	—	—	✓	✓	✓	✓	✓

✓: Available  
 —: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance “\_A”, “\_B”, or “\_C”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec (t<sub>GTISK</sub>).

**Table 19.15 Register settings for input/output pin function (PORT9)**

PSEL[4:0]settings	Function	Pin
		P905
00000b(value after reset)	Hi-Z/ JTAG/SWD	Hi-Z
00001b	AGT	—
00011b	GPT <sup>2</sup>	—
00100b	SCI <sup>1</sup>	CTS3_B
01001b	CLKOUT/ ACPHS/ RTC	—
10011b	USBFS	—
ASEL bit		—
ISEL bit		IRQ8
DSCR[1:0] bits	Drive capability control	L/M/H
NCODR bit	N-ch open- drain	✓
PCR bit	Pull-up	✓
144 pins product		✓
100 pins product		—

✓: Available

—: Setting prohibited

Note 1. Recommend using pins that have a letter appended to their names, for instance“\_A”, “\_B”, or “\_C”, to indicate group membership. For the interface, the AC portion of the electrical characteristics is measured for each group.

Note 2. There are 2 types output buffer which are middle drive and high drive. Recommend using same drive buffer for output skew spec ( $t_{GTISK}$ ).

## 20. Port Output Enable for GPT (POEG)

### 20.1 Overview

The Port Output Enable (POEG) function can place the General PWM Timer (GPT) output pins in the output disable state in one of the following ways:

- Input level detection of the GTETR<sub>Gn</sub> (n = A to D) pins
- Output-disable request from the GPT
- Comparator interrupt request detection
- Oscillation stop detection of the clock generation circuit
- Register settings

The GTETR<sub>Gn</sub> (n = A to D) pins can be used as GPT external trigger input pins.

Table 20.1 lists the POEG specifications, Figure 20.1 shows a block diagram, and Table 20.2 lists the input pins.

**Table 20.1 POEG specifications**

Parameter	Specifications
Output-disable control through input level detection	The GPT output pins can be disabled when a GTETR <sub>Gn</sub> rising edge or high level is sampled after polarity and filter selection.
Output-disable request from the GPT	When the GTIOCxA pin and the GTIOCxB pin are driven to an active level simultaneously, the GPT generates an output-disable request to the POEG. Through reception of these requests, the POEG can control whether the GTIOCxA and GTIOCxB pins are output-disabled.
Output-disable control through the comparator (ACMPHS) interrupt request detection	The GPT output pins can be disabled when an interrupt request is generated by a change in the output results of any of the comparators.
Output-disable control through oscillation stop detection	The GPT output pins can be disabled when oscillation of the clock generation circuit stops.
Output-disable control by software (registers)	The GPT output pins can be disabled by modifying the register settings.
Interrupt	Interrupts can be generated by detecting the input level of external trigger input pins (GTETR <sub>Gn</sub> pins). Interrupts can be generated when all GPT or ACMPHS output pins are driven to an active level simultaneously.
External trigger output to the GPT	The GTETR <sub>Gn</sub> signals can be output to the GPT after polarity and filter selection. (count start, count stop, count clear, up-count, down-count, or input capture function)
Noise filtering	For input from the GTETR <sub>Gn</sub> pins, PCLKB/1, PCLKB/8, PCLKB/32, or PCLKB/128 can be selected as the noise filtering clock. (Filtering is performed by sampling the input signals three times using the selected clock.) Positive or negative polarity can be selected for any of the GTETR <sub>Gn</sub> input pins. Signal state after polarity and filter selection can be monitored.
Module-stop function	Module-stop state can be set for each groups to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set for each groups.

Note: n = A to D, x = 0 to 5, 10 to 13

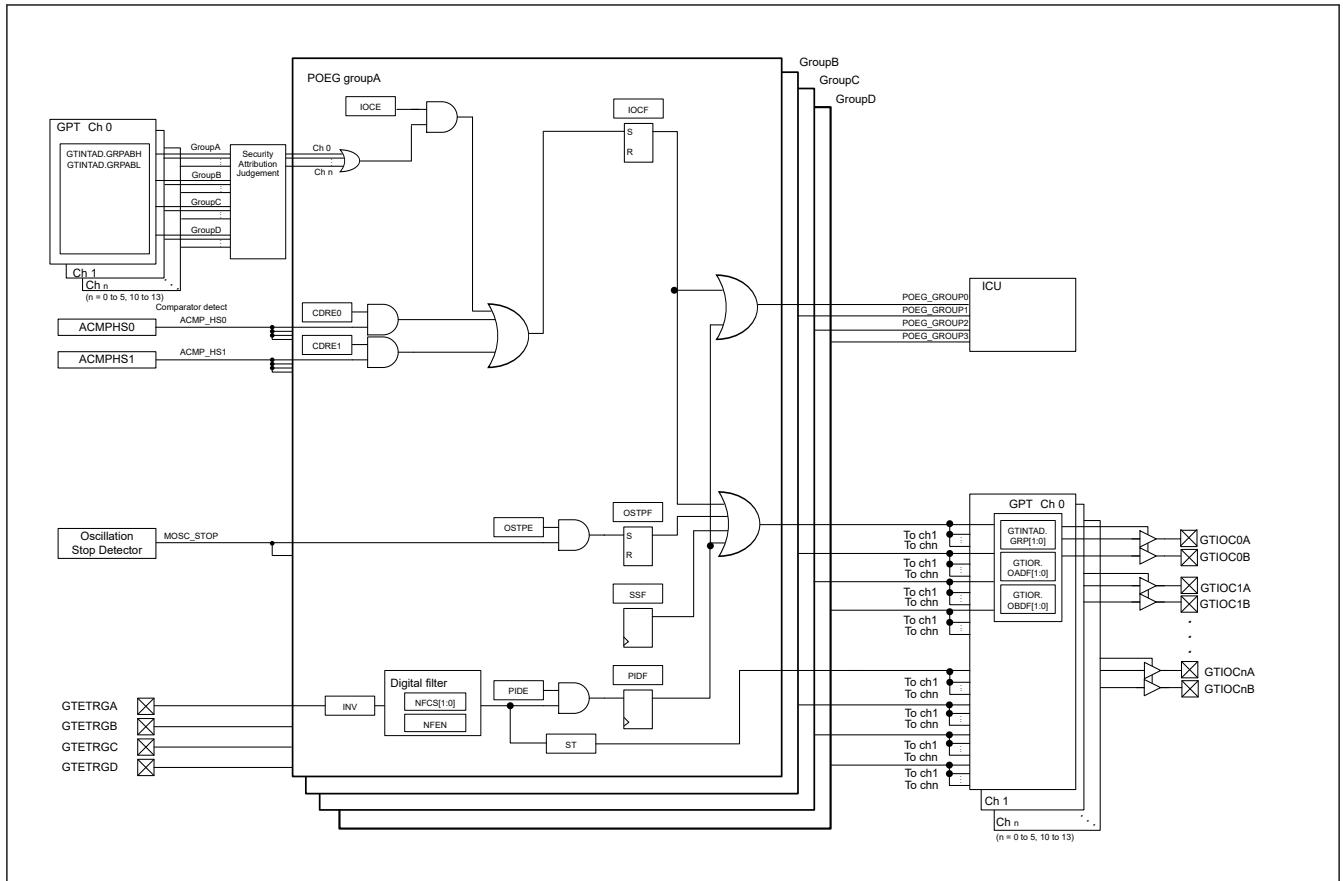


Table 20.2 POEG input pins

Pin name	I/O	Description
GTETRGA	Input	GPT output pin output-disable request signal or GPT external trigger input pin A
GTETRGB	Input	GPT output pin output-disable request signal or GPT external trigger input pin B
GTETRGC	Input	GPT output pin output-disable request signal or GPT external trigger input pin C
GTETRGD	Input	GPT output pin output-disable request signal or GPT external trigger input pin D

## 20.2 Register Descriptions

### 20.2.1 POEGn : POEG Group n Setting Register (n = A to D)

Base address: POEG = 0x4021\_2000  
 POEG\_NS = 0x5021\_2000

Offset address: 0x000 (POEGGA)  
 0x100 (POEGGB)  
 0x200 (POEGGC)  
 0x300 (POEGGD)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCFS[1:0]		NFEN	INV	—	—	—	—	—	—	—	—	—	—	—	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CDRE 1	CDRE 0	—	OSTP E	IOCE	PIDE	SSF	OSTP F	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIDF	Port Input Detection Flag 0: No output-disable request from the GTETRn pin occurred 1: Output-disable request from the GTETRn pin occurred.	R/W <sup>1</sup>
1	IOCF	Detection Flag for GPT or ACMPHS Output-Disable Request 0: No output-disable request from GPT or comparator interrupt occurred. 1: Output-disable request from GPT or comparator interrupt occurred.	R/W <sup>1</sup>
2	OSTPF	Oscillation Stop Detection Flag 0: No output-disable request from oscillation stop detection occurred 1: Output-disable request from oscillation stop detection occurred	R/W <sup>1</sup>
3	SSF	Software Stop Flag 0: No output-disable request from software occurred 1: Output-disable request from software occurred	R/W
4	PIDE	Port Input Detection Enable 0: Disable output-disable requests from the GTETRn pins 1: Enable output-disable requests from the GTETRn pins	R/W <sup>2</sup>
5	IOCE	Enable for GPT Output-Disable Request 0: Disable output-disable requests from GPT 1: Enable output-disable requests from GPT	R/W <sup>2</sup>
6	OSTPE	Oscillation Stop Detection Enable 0: Disable output-disable requests from oscillation stop detection 1: Enable output-disable requests from oscillation stop detection	R/W <sup>2</sup>
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CDRE0	ACMP_HS0 Enable 0: Disable output-disable request from ACMPHS0 1: Enable output-disable request from ACMPHS0	R/W <sup>2</sup>
9	CDRE1	ACMP_HS1 Enable 0: Disable output-disable request from ACMPHS1 1: Enable output-disable request from ACMPHS1	R/W <sup>2</sup>
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	ST	GTETRn Input Status Flag 0: GTETRn input after filtering was 0 1: GTETRn input after filtering was 1	R
27:17	—	These bits are read as 0. The write value should be 0.	R/W
28	INV	GTETRn Input Reverse 0: Input GTETRn as-is 1: Input GTETRn in reverse	R/W
29	NFEN	Noise Filter Enable 0: Disable noise filtering 1: Enable noise filtering	R/W
31:30	NFCS[1:0]	Noise Filter Clock Select 0 0: Sample GTETRn pin input level three times every PCLKB 0 1: Sample GTETRn pin input level three times every PCLKB/8 1 0: Sample GTETRn pin input level three times every PCLKB/32 1 1: Sample GTETRn pin input level three times every PCLKB/128	R/W

Note: S-TYPE3, P-TYPE3

Note 1. Only 0 can be written to clear the flag.

Note 2. Can be modified only once after a reset.

The POEGn (n = A to D) registers control the output-disable state of the GPT pins, interrupts, and the external trigger input to the GPT.

In the descriptions, POEGn represents the POEGn (n = A to D) registers.

### 20.3 Output-Disable Control Operation

If any of the following conditions is satisfied, the GTIOCxA, GTIOCxB, and the 3-phase PWM output for BLDC motor control pins can be set to output-disable:

- Input level or edge detection of the GTETR<sub>Gn</sub> pins  
When POEG<sub>Gn</sub>.PIDE is 1, the POEG<sub>Gn</sub>.PIDF flag is set to 1.
- Output-disable request from the GPT  
When POEG<sub>Gn</sub>.IOCE is 1, the POEG<sub>Gn</sub>.IOCF flag is set to 1 if the disable request is enabled by GTINTAD. The GTINTAD.GRPABH and GTINTAD.GRPABL settings apply to the group selected by the GPT register GTINTAD.GRP[1:0] or OPSCR.GRP[1:0].  
  
Note: The disable request is valid only when the security attributions of GPT and POEG are same. The disable request from GPT with a security attribution different from the security attribution of POEG is invalidated by the security attribution judgment.
- Comparator (ACMPHS) interrupt request detection  
Comparator interrupt detection is activated when any of the POEG<sub>Gn</sub>.CDRE<sub>i</sub> (i = 0, 1) is 1. When the associated comparator interrupt is generated, the GPT output pins are disabled. POEG<sub>Gn</sub>.IOCF indicates the detection status.
- Oscillation stop detection for the clock generation circuit  
While POEG<sub>Gn</sub>.OSTPE is 1, the halt status of the main clock oscillator is detected and the POEG<sub>Gn</sub>.OSTPF flag is set to 1.
- SSF bit setting  
When POEG<sub>Gn</sub>.SSF is set to 1, the GPT and PWM output are disabled.

The output-disable state is controlled in the GPT module. The output-disable of the GTIOC<sub>x</sub>A and GTIOC<sub>x</sub>B pins is set in the GTINTAD.GRP[1:0], GTIOR.OADF[1:0], and GTIOR.OBDF[1:0] bits in GPT<sub>x</sub>. The output-disable of the 3-phase PWM output for BLDC motor control pins is set in the OPSCR.GRP[1:0] bits and OPSCR.GODF bit in GPT\_OPS.

### 20.3.1 Pin Input Level Detection Operation

If the input conditions set in POEG<sub>Gn</sub>.PIDE, POEG<sub>Gn</sub>.NFCS[1:0], POEG<sub>Gn</sub>.NFEN, and POEG<sub>Gn</sub>.INV occur on the GTETR<sub>Gn</sub> pins, the GPT output pins are output-disabled.

#### 20.3.1.1 Digital Filter

Figure 20.2 shows high-level detection by the digital filter. When a high level associated with the POEG<sub>Gn</sub>.INV polarity setting is detected three times consecutively with the sampling clock selected in POEG<sub>Gn</sub>.NFCS[1:0] and POEG<sub>Gn</sub>.NFEN, the detected level is recognized as high, and the GPT output pins are output-disabled. If even one low level is detected during this interval, the detected level is not recognized as high. In addition, in an interval where the sampling clock is not output, changes of the levels on the GTETR<sub>Gn</sub> pins are ignored.

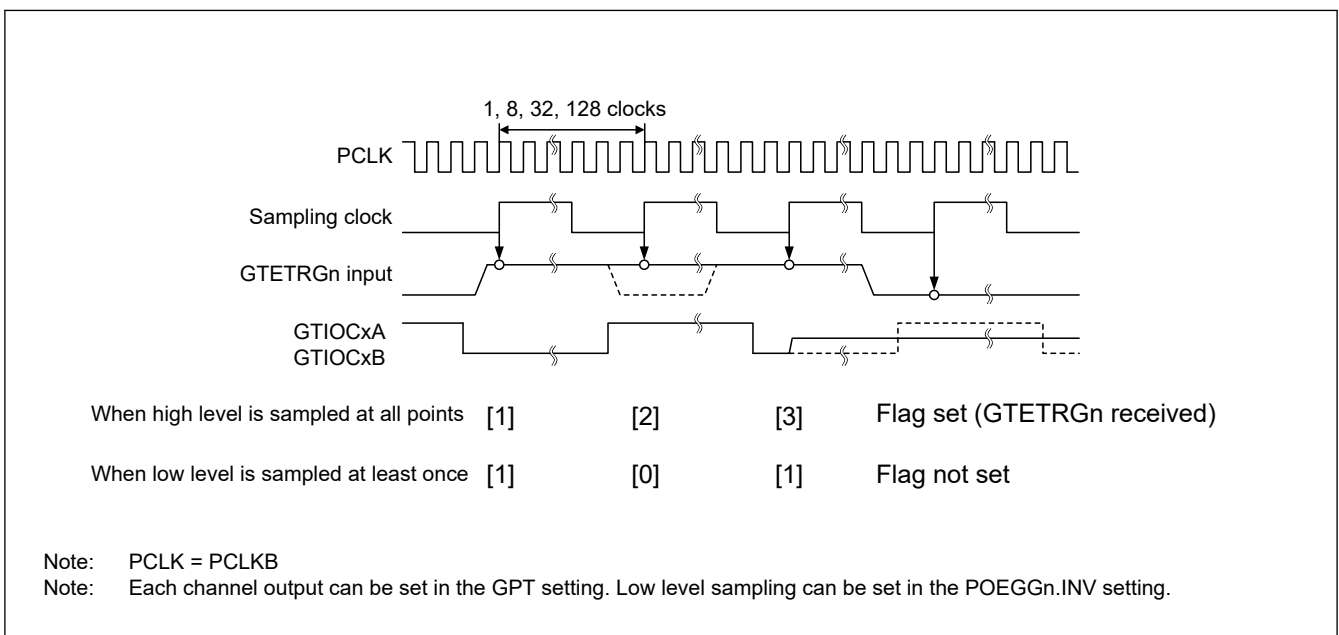


Figure 20.2 Example of digital filter operation

### 20.3.2 Output-Disable Requests from the GPT

For details on the operation, see the description for GTIOC Pin Output Negate Control in [section 21, General PWM Timer \(GPT\)](#).

### 20.3.3 Comparator Interrupt Detection

If POEGGn.CDREi (i = 0, 1) is 1 when an associated comparator interrupt request is generated, the GPT output pins are output-disabled for each group. The status flag is POEGGn.IOCF which is shared with GPT output-disable detection.

### 20.3.4 Output-Disable Control Using Detection of Stopped Oscillation

When the oscillation stop detection function in the clock generation circuit detects stopped oscillation while POEGGn.OSTPE is 1, the GPT output pins are output-disabled for each group.

### 20.3.5 Output-Disable Control Using Registers

The GPT output pins can be directly controlled by writing 1 to the Software Stop flag, POEGGn.SSF.

### 20.3.6 Release from Output-Disable

To release the GPT output pins placed in the output-disable state, either return them to their initial state with a reset or clear all of the following flags:

- POEGGn.PIDF
- POEGGn.IOCF
- POEGGn.OSTPF
- POEGGn.SSF

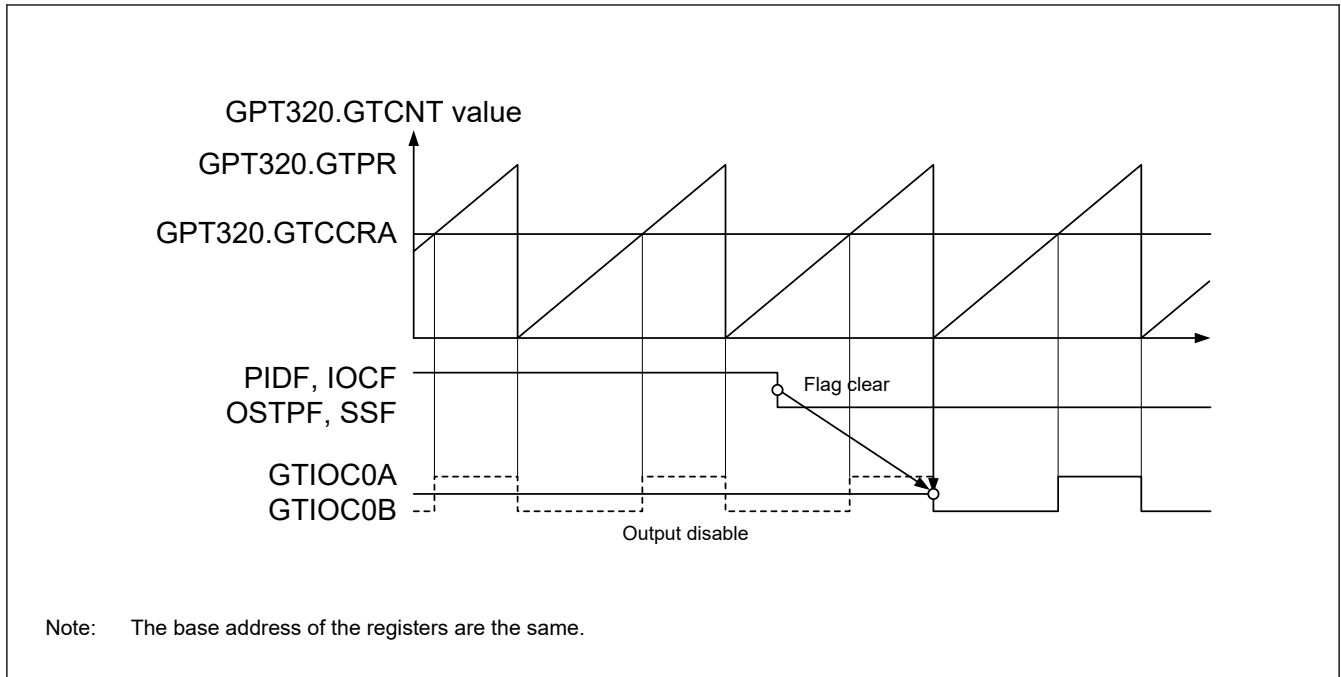
Writing 0 to the POEGGn.PIDF flag is ignored (the flag is not cleared) if the external input pins, GTETRGN are not disabled and the POEGGn.ST bit is not set to 0.

Writing 0 to the POEGGn.IOCF flag is valid (the flag is cleared) only if all of the GTST.OABHF and GTST.OABLF flags in the GPT are set to 0.

Writing 0 to the POEGGn.OSTPF flag is ignored (the flag is not cleared) if the OSTDSR.OSTDF flag in the clock generation circuit is not set to 0. In addition, when the flag set and release occur at the same time, the flag set takes precedence.

[Figure 20.3](#) shows the release timing for output-disable. The output-disable is released at the beginning of the next count cycle of the GPT after the flag is cleared.





**Figure 20.3** Output-disable release timing for GPT pin outputs

## 20.4 Interrupt Sources

The POEG generates an interrupt request for the following factors:

- Output-disable control by the input level detection
- Output-disable request from the GPT or the comparator (ACMPHS) interrupt request detection

Table 20.3 lists the conditions for interrupt requests.

**Table 20.3** Interrupt sources and conditions

Interrupt source	Symbol	Associated flag	Trigger conditions
POEG group A interrupt	POEG_GROUPA	POEGGA.IOCF	An output-disable request from a GPT disable request occurred
		POEGGA.PIDF	An output-disable request from the GTETRGA pin occurred
POEG group B interrupt	POEG_GROUPB	POEGGB.IOCF	An output-disable request from a GPT disable request occurred
		POEGGB.PIDF	An output-disable request from the GTETRGB pin occurred
POEG group C interrupt	POEG_GROUPC	POEGGC.IOCF	An output-disable request from a GPT disable request occurred
		POEGGC.PIDF	An output-disable request from the GTETRGC pin occurred
POEG group D interrupt	POEG_GROUPD	POEGGD.IOCF	An output-disable request from a GPT disable request occurred
		POEGGD.PIDF	An output-disable request from the GTETRGD pin occurred

## 20.5 External Trigger Output to the GPT

The POEG outputs signals generated by filtering and level detection of GTETRGN pins input signals as the GPT operation trigger signal for the following:

- Count start
- Count stop

- Count clear
- Up-count
- Down-count
- Input capture

For the POEGGn.INV polarity setting signal, when the same level is input three times continuously with the sampling clock selected in POEGGn.NFCS[1:0] and POEGGn.NFEN, that value is output. Set the control registers the same as for the input level detection operation described in section 20.3.1. Pin Input Level Detection Operation. The state after filtering can be monitored in POEGGn.ST.

Figure 20.4 shows the output timing of an external trigger to the GPT.

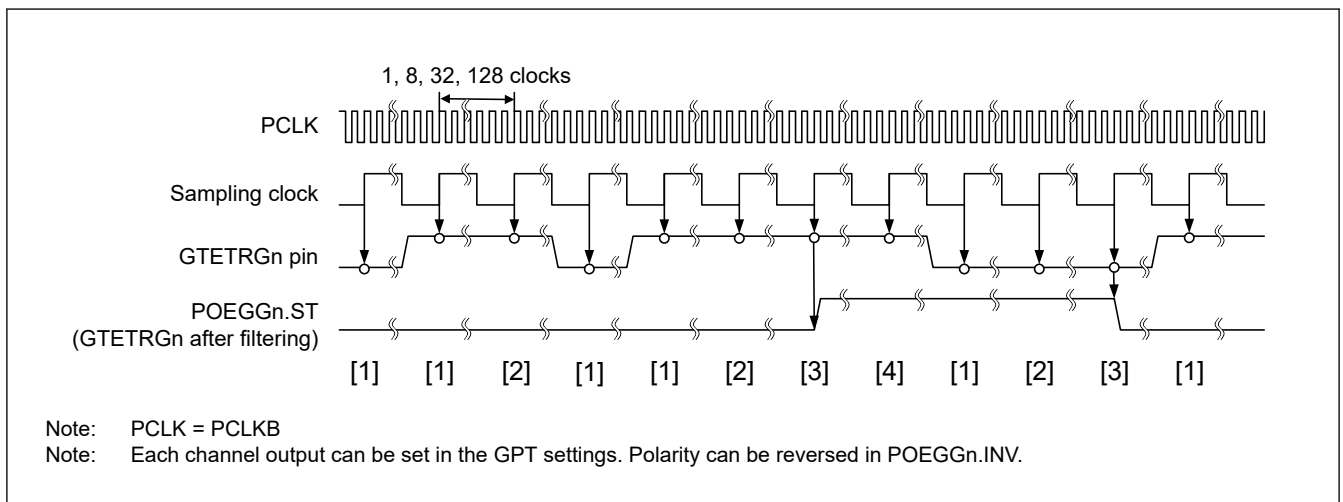


Figure 20.4 Output timing of external trigger to the GPT

## 20.6 Usage Notes

### 20.6.1 Transition to Software Standby Mode

When using the POEG, do not invoke Software Standby mode. In this mode, the POEG stops and therefore output disable of the pins cannot be controlled.

### 20.6.2 Specifying Pins Associated with the GPT

The POEG controls output-disable only when a pin is associated with the GPT in the PmnPFS.PMR and PmnPFS.PSEL settings. When the pin is specified as a general I/O pin, the POEG does not perform output-disable control.

## 21. General PWM Timer (GPT)

### 21.1 Overview

The General PWM Timer (GPT) is a 32-bit timer with  $GPT32 \times 6$  channels and a 16-bit timer with  $GPT16 \times 4$  channels. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. This GPT provides the A/D conversion start request function. The GPT can also be used as a general-purpose timer.

Table 21.1 lists the GPT specifications, Table 21.2 shows the GPT functions, and Figure 21.1 shows a block diagram.

**Table 21.1 GPT specifications**

Parameter	Description
Functions	<ul style="list-style-type: none"> <li>• 32 bits <math>\times</math> 6 channels (GPT32n (n = 0 to 5))</li> <li>• 16 bits <math>\times</math> 4 channels (GPT16m (m = 10 to 13))</li> <li>• Up-counting or down-counting (saw waves) or up/down-counting (triangle waves) for each counter</li> <li>• Clock sources independently selectable for each channel</li> <li>• Two input/output pins per channel</li> <li>• Two output compare/input capture registers per channel</li> <li>• For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use</li> <li>• In output compare operation, buffer switching can be at crests or troughs, enabling the generation of laterally asymmetric PWM waveforms</li> <li>• Registers for setting up frame cycles in each channel with capability for generating interrupts at overflow or underflow</li> <li>• Generation of dead times in PWM operation</li> <li>• Synchronous starting, stopping and clearing counters for arbitrary channels</li> <li>• Count start, count stop, count clear, up-count, down-count, or input capture operation in response to a maximum of 8 ELC events</li> <li>• Count start, count stop, count clear, up-count, down-count, or input capture operation in response to the status of two input pins</li> <li>• Count start, count stop, count clear, up-count, down-count, or input capture operation in response to a maximum of 4 external triggers</li> <li>• Control output disable request from the POEG</li> <li>• A/D conversion start request generation function</li> <li>• Compare match A to F event and overflow/underflow event can be output to the ELC</li> <li>• Enables the noise filter for input capture</li> <li>• Period count function</li> <li>• Logical operation between the channel output</li> <li>• Bus clock: PCLKA, Core clock: PCLKD</li> <li>• Frequency ratio: PCLKA:PCLKD = 1:N (N = 1/2/4/8/16/32/64)</li> </ul>

**Table 21.2 GPT functions (1 of 2)**

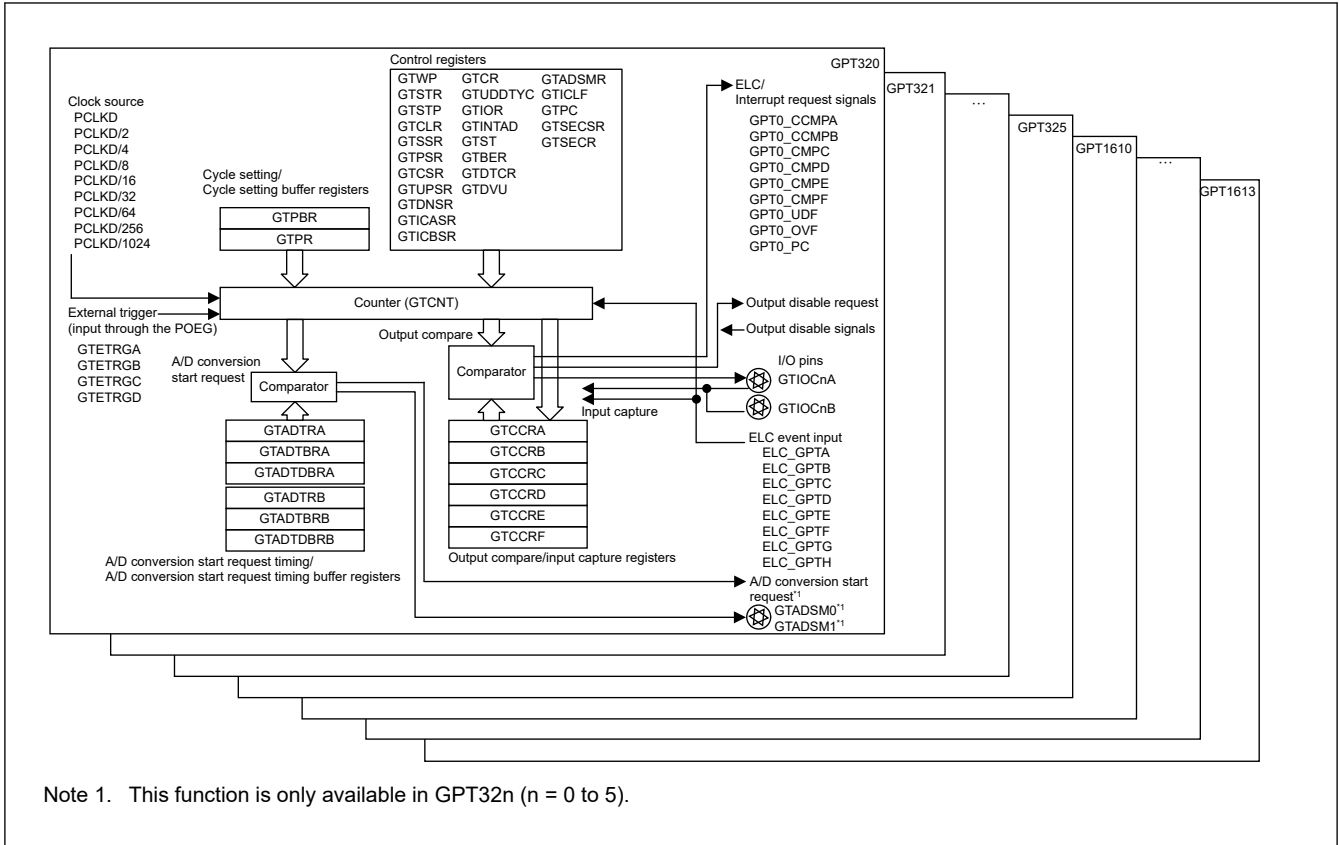
Parameter	Description
Count clock	PCLKD PCLKD/2 PCLKD/4 PCLKD/8 PCLKD/16 PCLKD/32 PCLKD/64 PCLKD/256 PCLKD/1024 GTETRGA, GTETRGB, GTETRGC, GTETRGD
Output compare/input capture registers (GTCCR)	GTCCRA GTCCRB
Compare/buffer registers	GTCCRC GTCCRD GTCCRE GTCCRF
Cycle setting buffer register	GTPBR
I/O pins	GTIOCnA GTIOCnB (n = 0 to 5, 10 to 13)

**Table 21.2 GPT functions (2 of 2)**

Parameter	Description	
External trigger input pin <sup>*1</sup>	GTETRGA GTETRGB GTETRGC GTETRGD	
Counter clear sources	GTPR register compare match Input capture Input pin status ELC event input GTETR Gn (n = A to D) pin input	
Period count function	Available (GPT32n (n = 0 to 3), GPT16m (m = 10))	
Compare match output	Low output	Available
	High output	Available
	Toggle output	Available
Input capture function	Available	
Automatic addition of dead time	Available (no dead time buffer)	
PWM mode	Available	
Phase count function	Available	
Buffer operation	Simultaneous operation disable control for multiple channels	
One-shot operation	Available	
DMAC/DTC activation	All the interrupt sources	
A/D conversion start request	Compare match of GTADTRA or GTADTRB register	
Interrupt sources	9 sources <ul style="list-style-type: none"> <li>• GTCCRA compare match/input capture(GPTn_CCMPA)</li> <li>• GTCCRB compare match/input capture(GPTn_CCMPB)</li> <li>• GTCCRC compare match(GPTn_CMPC)</li> <li>• GTCCRD compare match(GPTn_CMPD)</li> <li>• GTCCRE compare match(GPTn_CMPE)</li> <li>• GTCCRF compare match(GPTn_CMPF)</li> <li>• GTCNT overflow (GTPR compare match) (GPTn_OVF)</li> <li>• GTCNT underflow (GPTn_UDF)</li> <li>• GTPC count stop(GPTx_PC) (x = 0 to 3, 10)</li> </ul>	
Event linking (ELC) function	Available <sup>*2</sup>	
Noise filtering function	Available	
Logical operation between the channel output	Available	
Module-stop function	Module-stop state can be set for each channels to reduce power consumption	
TrustZone Filter	Security and Privilege attribution can be set for each channels	

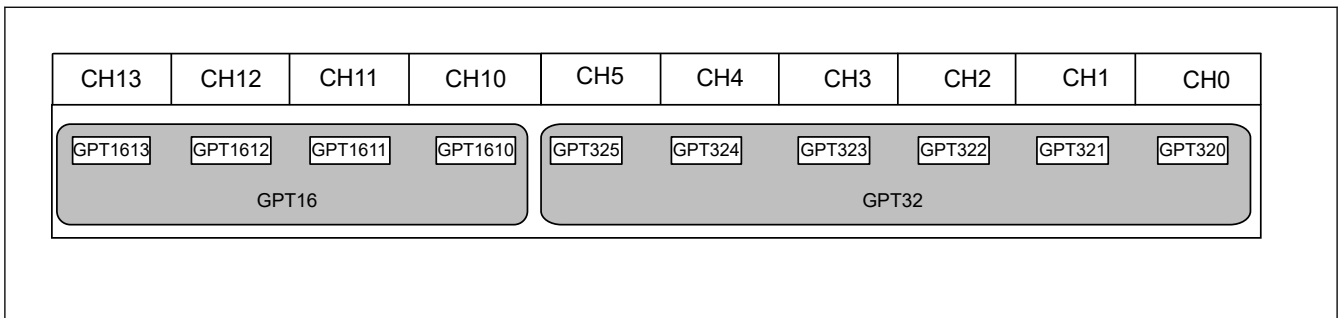
Note 1. GTETR Gn connects to GPT through the POEG module. Therefore, to use the GPT function, supply the POEG clock by clearing the MSTPCRD.MSTPDn (n = 11 to 14) bit.

Note 2. See [section 21.6. Operations Linked by ELC](#).



**Figure 21.1 GPT block diagram (Saw-wave PWM mode, Saw-wave one-shot pulse mode, Triangle-wave PWM mode 1,2,3)**

Figure 21.2 shows an example using multiple GPTs.



**Figure 21.2 Association between GPT channels and module names**

Table 21.3 lists the I/O pins.

**Table 21.3 GPT I/O pins**

Channel	Pin name	I/O	Function
Common	GTETRGr	Input	External trigger input pin r (input through the POEG)
	GTADSM0	Output	A/D conversion start request monitor 0 output pin
	GTADSM1	Output	A/D conversion start request monitor 1 output pin
GPT32n	GTIOCnA	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCnB	I/O	GTCCRB register input capture input/output compare output/PWM output pin
GPT16m	GTIOCmA	I/O	GTCCRA register input capture input/output compare output/PWM output pin
	GTIOCmB	I/O	GTCCRB register input capture input/output compare output/PWM output pin

Note: r: A to D  
n: 0 to 5

m: 10 to 13

## 21.2 Register Descriptions

### 21.2.1 GTWP : General PWM Timer Write-Protection Register

Base address:  $GPT32n = 0x4032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT32n\_NS = 0x5032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT16m = 0x4032\_2000 + 0x0100 \times m$  (m = 10 to 13)  
 $GPT16m\_NS = 0x5032\_2000 + 0x0100 \times m$  (m = 10 to 13)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	CMN WP	CLRWP	STPWP	STRWP	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WP	Register Write Disable 0: Write to the register enabled 1: Write to the register disabled	R/W
1	STRWP	GTSTR.CSTRT Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
2	STPWP	GTSTP.CSTOP Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
3	CLRWP	GTCLR.CCLR Bit Write Disable 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
4	CMNWP	Common Register Write Disabled 0: Write to the register is enabled 1: Write to the register is disabled	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	GTWP Key Code When 0xA5 is written to these bits, writing to the WP, STRWP, STPWP, CLRWP, and CMNWP bits are permitted. These bits are read as 0.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

GTWP enables or disables writing to registers to prevent accidental modification. Protection by the GTWP register is only for the writes by the CPU. GTWP does not protect registers from updates that occur in association with CPU writes.

#### WP bit (Register Write Disable)

The following is a list of write enabled or disabled registers:

GTSSR, GTPSR, GTCSSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTCNT, GTCORA, GTCORB, GTCORC, GTCORD, GTCORE, GTCORF, GTPR, GTPBR, GTADTRA, GTADTBRA, GTADTBRA, GTADTRB, GTADTRBR, GTADTRBR, GTDTCR, GTDVU, GTADSMR, GTICLF, GTPC.

**STRWP bit (GTSTR.CSTRT Bit Write Disable)**

The STRWP bit enables or disables starting the updating of counter values by writing to the CSTRTn bit (n = 0 to 5, 10 to 13) corresponding to a channel number in the GTSTR register.

The bit position of each CSTRTn bit in the GTSTR register is allocated to the channel with the corresponding number, and writing to the GTSTR register for any channel results in writing to the registers of all channels. The STRWP bit for each channel does not control writing but only controls updating of the CSTRT bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CSTRT bits of a channel for which the setting of the STRWP bit is 1 (disabling writing), the CSTRT bit for the given channel is not updated, but the CSTRT bits corresponding to channel for which the setting of the STRWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT320.GTWP.STRWP bit is 0 (enabling writing), writing 1 to the GPT321.GTSTR.CSTRT0 bit when its current setting is 0 causes the value to be updated, and the GPT320.GTCNT counter starts to run. When the setting of the GPT320.GTWP.STRWP bit is 1 (disabling writing), writing 1 to the GPT321.GTSTR.CSTRT0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT320.GTCNT counter does not run.

If you want to protect all bits in the GTSTR register from being updated, set the STRWP bits of all channels to 1.

**STPWP bit (GTSTP.CSTOP Bit Write Disable)**

The STPWP bit enables or disables starting the updating of counter values by writing to the CSTOPn bit (n = 0 to 5, 10 to 13) corresponding to a channel number in the GTSTP register.

The bit position of each CSTOPn bit in the GTSTP registers is allocated to the channel with the corresponding number, and the writing to the GTSTP register for any channel results in writing to the registers of all channels. The STPWP bit for each channel does not control writing but only controls updating of the CSTOP bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CSTOP bits of a channel for which the setting of the STPWP bit is 1 (disabling writing), the CSTOP bit for the given channel is not updated, but the CSTOP bits corresponding to channel for which the setting of the STPWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT320.GTWP.STPWP bit is 0 (enabling writing), writing 1 to the GPT321.GTSTP.CSTOP0 bit when its current setting is 0 causes the value to be updated, and the GPT320.GTCNT counter is stopped. When the setting of the GPT320.GTWP.STPWP bit is 1 (disabling writing), writing 1 to the GPT321.GTSTP.CSTOP0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT320.GTCNT counter is not stopped.

If you want to protect all bits in the GTSTP register from being updated, set the STPWP bits of all channels to 1.

**CLRWP bit (GTCLR.CCLR Bit Write Disable)**

CLRWP bit enables or disables starting the updating of counter values by writing to the CCLRn bit (n = 0 to 5, 10 to 13) corresponding to a channel number in the GTCLR register.

The bit position of each CCLRn bit in the GTCLR registers is allocated to the channel with the corresponding number, and the writing to the GTCLR register for any channel results in writing to the registers of all channels. The CLRWP bit for each channel does not control writing but only controls updating of the CCLR bit for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the CCLR bits of a channel for which the setting of the CLRWP bit is 1 (disabling writing), the CCLR bit for the given channel is not updated, but the CCLR bits corresponding to channel for which the setting of the CLRWP bit is 0 (enabling writing) are updated. For example, when the setting of the GPT320.GTWP.CLRWP bit is 0 (enabling writing), writing 1 to the GPT321.GTCLR.CCLR0 bit when its current setting is 0 causes the value to be updated, and the GPT320.GTCNT counter is cleared. When the setting of the GPT320.GTWP.CLRWP bit is 1 (disabling writing), writing 1 to the GPT321.GTCLR.CCLR0 bit when its current setting is 0 leaves the bit with the value 0, and the GPT320.GTCNT counter is not cleared.

If you want to protect all bits in the GTCLR register from being updated, set the CLRWP bits of all channels to 1.

**CMNWP bit (Common Register Write Disabled)**

CMNWP bit enables or disables starting the updating of counter values by writing to the SECSELn bit (n = 0 to 5, 10 to 13) corresponding to a channel number in the GTSECSR register or to the GTSECR register.

The bit position of each SECSEL bit in the GTSECSR registers is allocated to the channel with the corresponding number, and the writing to the GTSECSR register for any channel results in writing to the registers of all channels. Writing to the

GTSECR register of any channel leads to writing to the registers of all channels. The CMNWP bit for each channel does not control writing but only controls updating of the SECSEL bit and the GTSECR register value for the corresponding channel when simultaneously writing to all channels.

Therefore, when writing to the SECSEL bit and the GTSECR register value of a channel for which the setting of the CMNWP bit is 1 (disabling writing), the SECSEL bit and the GTSECR register value for the given channel is not updated, but the SECSEL bit and the GTSECR register value corresponding to channel for which the setting of the CMNWP bit is 0 (enabling writing) are updated.

For example, when the setting of the GPT320.GTWP.CMNWP bit is 0 (enabling writing), writing to the GPT321.GTSECSR.SECSEL0 bit causes the value of the GPT320.GTSECSR.SECSEL0 bit to be updated. In the same way, writing to the GPT321.GTSECR register updates the value of the GPT320.GTSECR register. When the setting of the GPT320.GTWP.CMNWP bit is 1 (disabling writing), writing to the GPT321.GTSECSR.SECSEL0 bit does not cause the value of the GPT320.GTSECSR.SECSEL0 bit to be updated. In the same way, writing to the GPT321.GTSECR register does not update the value of the GPT320.GTSECR register.

If you want to protect all bits in the GTSECSR and GTSECR registers from being updated, set the CMNWP bits of all channels to 1.

**PRKEY[7:0] bit (GTWP Key Code)**

This bit controls whether the WP, STRWP, STPWP, CLRWP, and CMNWP bits can be overwritten.

**21.2.2 GTSTR : General PWM Timer Software Start Register**

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CSTR T13	CSTR T12	CSTR T11	CSTR T10	—	—	—	—	CSTR T5	CSTR T4	CSTR T3	CSTR T2	CSTR T1	CSTR T0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CSTR0 to CSTR5	Channel n GTCNT Count Start (n : the same as bit position value) 0: GTCNT counter not start 1: GTCNT counter start	R/W
9:6	—	These bits are read as 0. The write value should be 0.	R/W
13:10	CSTR10 to CSTR13	Channel n GTCNT Count Start (n : the same as bit position value) 0: GTCNT counter not start 1: GTCNT counter start	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The GTSTR starts the GTCNT counter operation for each channel n, where n = 0 to 5, 10 to 13.

The GTSTR bit number represents the channel number. The GTSTR register of each channel is shared by all of the channels. The GTCNT counter starts for the channel associated with the GTSTR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter and the value of GTSTR register.

For the association between module names and channel numbers, see [Figure 21.2](#).

Contrary to a security attribute for write or read access, if the security and privilege attributes set on each channel has the security violation or the privilege violation, the bit of the channel number that violates security cannot be written or read, and the read value of the bit is "0".



**CSTRn bits (Channel n GTCNT Count Start (n =0 to 5, 10 to 13))**

The CSTRn bits start channel n of the GTCNT counter operation. Writing to the GTSTR.CSTRn bit (n = 0 to 5, 10 to 13) has no effect unless the GTSSR.CSTRT bit is set to 1.

The read data shows the counter status of each channel (GTCR.CST bit). A value of 0 means the counter is stopped and 1 means the counter is running.

**21.2.3 GTSTP : General PWM Timer Software Stop Register**

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CSTO P13	CSTO P12	CSTO P11	CSTO P10	—	—	—	—	CSTO P5	CSTO P4	CSTO P3	CSTO P2	CSTO P1	CSTO P0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
5:0	CSTOP0 to CSTOP5	Channel n GTCNT Count Stop (n : the same as bit position value) 0: GTCNT counter not stop 1: GTCNT counter stop	R/W
9:6	—	These bits are read as 0. The write value should be 0.	R/W
13:10	CSTOP10 to CSTOP13	Channel n GTCNT Count Stop (n : the same as bit position value) 0: GTCNT counter not stop 1: GTCNT counter stop	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The GTSTP stops the GTCNT counter operation for each channel n, where n = 0 to 5, 10 to 13.

The GTSTP bit number represents the channel number. The GTSTP register of each channel is shared by all the channels. The GTCNT counter stops for the channel associated with the GTSTP bit number where 1 is written. Writing 0 has no effect on the status of the GTCNT counter and the value of GTSTP register.

For the association between module names and channel numbers, see [Figure 21.2](#).

Contrary to a security attribute for write or read access, if the security and privilege attributes set on each channel has the security violation or the privilege violation, the bit of the channel number that violates security cannot be written or read, and the read value of the bit is "0".

**CSTOPn bits (Channel n GTCNT Count Stop (n =0 to 5, 10 to 13))**

The CSTOPn bits stop channel n of the GTCNT counter operation. Writing to the GTSTP.CSTOPn bit (n = 0 to 5, 10 to 13) has no effect unless the GTPSR.CSTOP bit is set to 1. The read data shows the counter status of each channel (invert of GTCR.CST bit). A value of 0 means the counter is running and 1 means the counter is stopped.

### 21.2.4 GTCLR : General PWM Timer Software Clear Register

Base address:  $GPT32n = 0x4032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT32n\_NS = 0x5032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT16m = 0x4032\_2000 + 0x0100 \times m$  (m = 10 to 13)  
 $GPT16m\_NS = 0x5032\_2000 + 0x0100 \times m$  (m = 10 to 13)

Offset address: 0x0C

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

Bit	Symbol	Function	R/W
5:0	CCLR0 to CCLR5	Channel n GTCNT Count Clear (n : the same as bit position value) 0: GTCNT counter is not cleared 1: GTCNT counter is cleared	W
9:6	—	These bits are read as 0. The write value should be 0.	W
13:10	CCLR10 to CCLR13	Channel n GTCNT Count Clear (n : the same as bit position value) 0: GTCNT counter is not cleared 1: GTCNT counter is cleared	W
31:14	—	These bits are read as 0. The write value should be 0.	W

The GTCLR is a write-only register that clears the GTCNT counter operation for each channel n, where n = 0 to 5, 10 to 13. The GTCLR bit number represents the channel number. The GTCLR register of each channel is shared by all the channels. The GTCNT counter is cleared for the channel associated with the GTCLR bit number where 1 is written. Writing 0 has no effect on the status of GTCNT counter.

For the association between module names and channel numbers, see [Figure 21.2](#).

Contrary to a security attribute for write or read access, if the security and privilege attributes set on each channel has the security violation or the privilege violation, the bit of the channel number that violates security cannot be written or read, and the read value of the bit is "0".

#### CCLRn bits (Channel n GTCNT Count Clear (n = 0 to 5, 10 to 13))

When the counting direction flag is set for decrement (GTST.TUCF flag = 0) with saw-wave mode selected in the GTCR.MD[2:0] bits, the value of the GTCNT counter becomes that of the corresponding GTPR register in response to writing 1 to the CCLRn bit. The value of the counter becomes 0x0000 0000 with other settings. These bits are read as 0.

### 21.2.5 GTSSR : General PWM Timer Start Source Select Register

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTR T	—	—	—	—	—	—	—	SSEL CH	SSEL CG	SSEL CF	SSEL CE	SSEL CD	SSEL CC	SSEL CB	SSEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SSCB FAH	SSCB FAL	SSCB RAH	SSCB RAL	SSCA FBH	SSCA FBL	SSCA RBH	SSCA RBL	SSGT RGDF	SSGT RGDR	SSGT RGCF	SSGT RGCR	SSGT RBF	SSGT RBR	SSGT RGAF	SSGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSGTRGAR	GTETRGA Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGA input 1: Counter start enabled on the rising edge of GTETRGA input	R/W
1	SSGTRGAF	GTETRGA Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGA input 1: Counter start enabled on the falling edge of GTETRGA input	R/W
2	SSGTRGBR	GTETRGB Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGB input 1: Counter start enabled on the rising edge of GTETRGB input	R/W
3	SSGTRGBF	GTETRGB Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGB input 1: Counter start enabled on the falling edge of GTETRGB input	R/W
4	SSGTRGCR	GTETRGC Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGC input 1: Counter start enabled on the rising edge of GTETRGC input	R/W
5	SSGTRGCF	GTETRGC Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGC input 1: Counter start enabled on the falling edge of GTETRGC input	R/W
6	SSGTRGDR	GTETRGD Pin Rising Input Source Counter Start Enable 0: Counter start disabled on the rising edge of GTETRGD input 1: Counter start enabled on the rising edge of GTETRGD input	R/W
7	SSGTRGDF	GTETRGD Pin Falling Input Source Counter Start Enable 0: Counter start disabled on the falling edge of GTETRGD input 1: Counter start enabled on the falling edge of GTETRGD input	R/W
8	SSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter start enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	SSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter start enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	SSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter start enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W

Bit	Symbol	Function	R/W
11	SSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter start enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	SSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter start enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	SSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Start Enable 0: Counter start disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter start enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	SSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter start enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	SSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Start Enable 0: Counter start disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter start enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	SSELCA	ELC_GPTA Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTA input 1: Counter start enabled at the ELC_GPTA input	R/W
17	SSELCB	ELC_GPTB Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTB input 1: Counter start enabled at the ELC_GPTB input	R/W
18	SSELCC	ELC_GPTC Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTC input 1: Counter start enabled at the ELC_GPTC input	R/W
19	SSELCD	ELC_GPTD Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTD input 1: Counter start enabled at the ELC_GPTD input	R/W
20	SSELCE	ELC_GPTE Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTE input 1: Counter start enabled at the ELC_GPTE input	R/W
21	SSELCF	ELC_GPTF Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTF input 1: Counter start enabled at the ELC_GPTF input	R/W
22	SSELCG	ELC_GPTG Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTG input 1: Counter start enabled at the ELC_GPTG input	R/W
23	SSELCH	ELC_GPTH Event Source Counter Start Enable 0: Counter start disabled at the ELC_GPTH input 1: Counter start enabled at the ELC_GPTH input	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTRT	Software Source Counter Start Enable 0: Counter start disabled by the GTSTR register 1: Counter start enabled by the GTSTR register	R/W

Note: n = 0 to 5, 10 to 13

The GTSSR sets the source to start the GTCNT counter.

Input from GTETR Gn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

**SSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Start Enable)**

The SSGTRGAR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGA pin input.

**SSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Start Enable)**

The SSGTRGAF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGA pin input.

**SSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Start Enable)**

The SSGTRGBR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGB pin input.

**SSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Start Enable)**

The SSGTRGBF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGB pin input.

**SSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Start Enable)**

The SSGTRGCR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGC pin input.

**SSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Start Enable)**

The SSGTRGCF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGC pin input.

**SSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Start Enable)**

The SSGTRGDR bit enables or disables the GTCNT counter start on the rising edge of the GTETRGD pin input.

**SSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Start Enable)**

The SSGTRGDF bit enables or disables the GTCNT counter start on the falling edge of the GTETRGD pin input.

**SSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Start Enable)**

The SSCARBL bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

**SSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Start Enable)**

The SSCARBH bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

**SSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Start Enable)**

The SSCAFBL bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

**SSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Start Enable)**

The SSCAFBH bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

**SSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Start Enable)**

The SSCBRAL bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

**SSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Start Enable)**

The SSCBRAH bit enables or disables the GTCNT counter start on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

**SSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Start Enable)**

The SSCBFAL bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

**SSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Start Enable)**

The SSCBFAH bit enables or disables the GTCNT counter start on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

**SSELCm bit (ELC\_GPTm Event Source Counter Start Enable) (m = A to H)**

The SSELCm bit enables or disables the GTCNT counter start at the ELC\_GPTm event input.

**CSTRT bit (Software Source Counter Start Enable)**

The CSTRT bit enables or disables the GTCNT counter start by GTSTR register.

**21.2.6 GTPSR : General PWM Timer Stop Source Select Register**

Base address:  $GPT32n = 0x4032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT32n\_NS = 0x5032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT16m = 0x4032\_2000 + 0x0100 \times m$  (m = 10 to 13)  
 $GPT16m\_NS = 0x5032\_2000 + 0x0100 \times m$  (m = 10 to 13)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTO P	—	—	—	—	—	—	—	PSEL CH	PSEL CG	PSEL CF	PSEL CE	PSEL CD	PSEL CC	PSEL CB	PSEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSCB FAH	PSCB FAL	PSCB RAH	PSCB RAL	PSCA FBH	PSCA FBL	PSCA RBH	PSCA RBL	PSGT RGDF	PSGT RGDR	PSGT RGCF	PSGT RGCR	PSGT RGBF	PSGT RGBR	PSGT RGAF	PSGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PSGTRGAR	GTETRGA Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGA input 1: Counter stop enabled on the rising edge of GTETRGA input	R/W
1	PSGTRGAF	GTETRGA Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGA input 1: Counter stop enabled on the falling edge of GTETRGA input	R/W
2	PSGTRGBR	GTETRGB Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGB input 1: Counter stop enabled on the rising edge of GTETRGB input	R/W
3	PSGTRGBF	GTETRGB Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGB input 1: Counter stop enabled on the falling edge of GTETRGB input	R/W
4	PSGTRGCR	GTETRGC Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGC input 1: Counter stop enabled on the rising edge of GTETRGC input	R/W
5	PSGTRGCF	GTETRGC Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGC input 1: Counter stop enabled on the falling edge of GTETRGC input	R/W
6	PSGTRGDR	GTETRGD Pin Rising Input Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTETRGD input 1: Counter stop enabled on the rising edge of GTETRGD input	R/W
7	PSGTRGDF	GTETRGD Pin Falling Input Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTETRGD input 1: Counter stop enabled on the falling edge of GTETRGD input	R/W
8	PSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter stop enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W

Bit	Symbol	Function	R/W
9	PSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter stop enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	PSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter stop enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	PSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter stop enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	PSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter stop enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	PSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Stop Enable 0: Counter stop disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter stop enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	PSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter stop enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	PSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Stop Enable 0: Counter stop disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter stop enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	PSELCA	ELC_GPTA Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTA input 1: Counter stop enabled at the ELC_GPTA input	R/W
17	PSELCB	ELC_GPTB Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTB input 1: Counter stop enabled at the ELC_GPTB input	R/W
18	PSELCC	ELC_GPTC Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTC input 1: Counter stop enabled at the ELC_GPTC input	R/W
19	PSELCD	ELC_GPTD Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTD input 1: Counter stop enabled at the ELC_GPTD input	R/W
20	PSELCE	ELC_GPTE Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTE input 1: Counter stop enabled at the ELC_GPTE input	R/W
21	PSELCF	ELC_GPTF Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTF input 1: Counter stop enabled at the ELC_GPTF input	R/W
22	PSELCG	ELC_GPTG Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTG input 1: Counter stop enabled at the ELC_GPTG input	R/W
23	PSELCH	ELC_GPTH Event Source Counter Stop Enable 0: Counter stop disabled at the ELC_GPTH input 1: Counter stop enabled at the ELC_GPTH input	R/W

Bit	Symbol	Function	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTOP	Software Source Counter Stop Enable 0: Counter stop disabled by the GTSTP register 1: Counter stop enabled by the GTSTP register	R/W

Note: n = 0 to 5, 10 to 13

The GTPSR sets the source to stop the GTCNT counter.

Inputs from GTETR<sub>Gn</sub> (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

#### **PSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Stop Enable)**

The PSGTRGAR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGA pin input.

#### **PSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Stop Enable)**

The PSGTRGAF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGA pin input.

#### **PSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Stop Enable)**

PSGTRGBR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGB pin input.

#### **PSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Stop Enable)**

The PSGTRGBF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGB pin input.

#### **PSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Stop Enable)**

PSGTRGCR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGC pin input.

#### **PSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Stop Enable)**

The PSGTRGCF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGC pin input.

#### **PSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Stop Enable)**

PSGTRGDR bit enables or disables the GTCNT counter stop on the rising edge of the GTETRGD pin input.

#### **PSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Stop Enable)**

The PSGTRGDF bit enables or disables the GTCNT counter stop on the falling edge of the GTETRGD pin input.

#### **PSCARBL bit (GTIOC<sub>nA</sub> Pin Rising Input during GTIOC<sub>nB</sub> Value Low Source Counter Stop Enable)**

The PSCARBL bit enables or disables the GTCNT counter stop on the rising edge of the GTIOC<sub>nA</sub> pin input, when GTIOC<sub>nB</sub> input is 0.

#### **PSCARBH bit (GTIOC<sub>nA</sub> Pin Rising Input during GTIOC<sub>nB</sub> Value High Source Counter Stop Enable)**

The PSCARBH bit enables or disables the GTCNT counter stop on the rising edge of the GTIOC<sub>nA</sub> pin input, when GTIOC<sub>nB</sub> input is 1.

#### **PSCAFBL bit (GTIOC<sub>nA</sub> Pin Falling Input during GTIOC<sub>nB</sub> Value Low Source Counter Stop Enable)**

The PSCAFBL bit enables or disables the GTCNT counter stop on the falling edge of the GTIOC<sub>nA</sub> pin input, when GTIOC<sub>nB</sub> input is 0.

#### **PSCAFBH bit (GTIOC<sub>nA</sub> Pin Falling Input during GTIOC<sub>nB</sub> Value High Source Counter Stop Enable)**

The PSCAFBH bit enables or disables the GTCNT counter stop on the falling edge of the GTIOC<sub>nA</sub> pin input, when GTIOC<sub>nB</sub> input is 1.

#### **PSCBRAL bit (GTIOC<sub>nB</sub> Pin Rising Input during GTIOC<sub>nA</sub> Value Low Source Counter Stop Enable)**

The PSCBRAL bit enables or disables the GTCNT counter stop on the rising edge of the GTIOC<sub>nB</sub> pin input, when GTIOC<sub>nA</sub> input is 0.



**PSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Stop Enable)**

The PSCBRAH bit enables or disables the GTCNT counter stop on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

**PSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Stop Enable)**

The PSCBFAL bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

**PSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Stop Enable)**

The PSCBFAH bit enables or disables the GTCNT counter stop on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

**PSELCm bit (ELCm Event Source Counter Stop Enable) (m = A to H)**

The PSELCm bit enables or disables the GTCNT counter stop at the ELC\_GPTm event input.

**CSTOP bit (Software Source Counter Stop Enable)**

The CSTOP bit enables or disables the GTCNT counter stop by the GTSTP register.

**21.2.7 GTCSR : General PWM Timer Clear Source Select Register**

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CCLR	—	—	—	—	—	—	—	CSEL CH	CSEL CG	CSEL CF	CSEL CE	CSEL CD	CSEL CC	CSEL CB	CSEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CSCB FAH	CSCB FAL	CSCB RAH	CSCB RAL	CSCA FBH	CSCA FBL	CSCA RBH	CSCA RBL	CSGT RGDF	CSGT RGDR	CSGT RGCF	CSGT RGCR	CSGT RGBF	CSGT RGBR	CSGT RGAF	CSGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CSGTRGAR	GTETRGA Pin Rising Input Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTETRGA input 1: Counter clear enabled on the rising edge of GTETRGA input	R/W
1	CSGTRGAF	GTETRGA Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGA input 1: Counter clear enabled on the falling edge of GTETRGA input	R/W
2	CSGTRGBR	GTETRGB Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGB input 1: Enable counter clear on the rising edge of GTETRGB input	R/W
3	CSGTRGBF	GTETRGB Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGB input 1: Counter clear enabled on the falling edge of GTETRGB input	R/W
4	CSGTRGCR	GTETRGC Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGC input 1: Enable counter clear on the rising edge of GTETRGC input	R/W
5	CSGTRGCF	GTETRGC Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGC input 1: Counter clear enabled on the falling edge of GTETRGC input	R/W

Bit	Symbol	Function	R/W
6	CSGTRGDR	GTETRGD Pin Rising Input Source Counter Clear Enable 0: Disable counter clear on the rising edge of GTETRGD input 1: Enable counter clear on the rising edge of GTETRGD input	R/W
7	CSGTRGDF	GTETRGD Pin Falling Input Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTETRGD input 1: Counter clear enabled on the falling edge of GTETRGD input	R/W
8	CSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter clear enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	CSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter clear enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	CSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter clear enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	CSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter clear enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	CSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter clear enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	CSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Clear Enable 0: Counter clear disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter clear enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	CSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter clear enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	CSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Clear Enable 0: Counter clear disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter clear enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	CSELCA	ELC_GPTA Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTA input 1: Counter clear enabled at the ELC_GPTA input	R/W
17	CSELCB	ELC_GPTB Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTB input 1: Counter clear enabled at the ELC_GPTB input	R/W
18	CSELCC	ELC_GPTC Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTC input 1: Counter clear enabled at the ELC_GPTC input	R/W
19	CSELCD	ELC_GPTD Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTD input 1: Counter clear enabled at the ELC_GPTD input	R/W

Bit	Symbol	Function	R/W
20	CSELCE	ELC_GPTE Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTE input 1: Counter clear enabled at the ELC_GPTE input	R/W
21	CSELCF	ELC_GPTF Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTF input 1: Counter clear enabled at the ELC_GPTF input	R/W
22	CSELG	ELC_GPTG Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTG input 1: Counter clear enabled at the ELC_GPTG input	R/W
23	CSELCH	ELC_GPTH Event Source Counter Clear Enable 0: Counter clear disabled at the ELC_GPTH input 1: Counter clear enabled at the ELC_GPTH input	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CCLR	Software Source Counter Clear Enable 0: Counter clear disabled by the GTCLR register 1: Counter clear enabled by the GTCLR register	R/W

Note: n = 0 to 5, 10 to 13

The GTCSCR sets the source to clear the GTCNT counter.

Counter clearing can be executed whether the counter is running (GTCR.CST = 1) or stopped (GTCR.CST = 0).

Inputs from GTETRGN (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

#### **CSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Clear Enable)**

The CSGTRGAR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGA pin input.

#### **CSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Clear Enable)**

The CSGTRGAF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGA pin input.

#### **CSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Clear Enable)**

The CSGTRGBR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGB pin input.

#### **CSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Clear Enable)**

The CSGTRGBF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGB pin input.

#### **CSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Clear Enable)**

The CSGTRGCR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGC pin input.

#### **CSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Clear Enable)**

The CSGTRGCF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGC pin input.

#### **CSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Clear Enable)**

The CSGTRGDR bit enables or disables the GTCNT counter clear on the rising edge of the GTETRGD pin input.

#### **CSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Clear Enable)**

The CSGTRGDF bit enables or disables the GTCNT counter clear on the falling edge of the GTETRGD pin input.

#### **CSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Clear Enable)**

The CSCARBL bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

#### **CSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Clear Enable)**

The CSCARBH bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

**CSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Clear Enable)**

The CSCAFBL bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

**CSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Clear Enable)**

The CSCAFBH bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

**CSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Clear Enable)**

The CSCBRAL bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

**CSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Clear Enable)**

The CSCBRAH bit enables or disables the GTCNT counter clear on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

**CSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Clear Enable)**

The CSCBFAL bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

**CSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Clear Enable)**

The CSCBFAH bit enables or disables the GTCNT counter clear on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

**CSELCm bit (ELCm Event Source Counter Clear Enable) (m = A to H)**

The CSELCm bit enables or disables the GTCNT counter clear at the ELC\_GPTm event input.

**CCLR bit (Software Source Counter Clear Enable)**

The CCLR bit enables or disables the GTCNT counter clear by the GTCLR register.

**21.2.8 GTUPSR : General PWM Timer Up Count Source Select Register**

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	USEL CH	USEL CG	USEL CF	USEL CE	USEL CD	USEL CC	USEL CB	USEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	USCB FAH	USCB FAL	USCB RAH	USCB RAL	USCA FBH	USCA FBL	USCA RBH	USCA RBL	USGT RGDF	USGT RGDR	USGT RGCF	USGT RGCR	USGT RGBF	USGT RGBR	USGT RGAF	USGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USGTRGAR	GTETRGA Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGA input 1: Counter count up enabled on the rising edge of GTETRGA input	R/W
1	USGTRGAF	GTETRGA Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGA input 1: Counter count up enabled on the falling edge of GTETRGA input	R/W

Bit	Symbol	Function	R/W
2	USGTRGBR	GTETRGRB Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGRB input 1: Counter count up enabled on the rising edge of GTETRGRB input	R/W
3	USGTRGBF	GTETRGRB Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGRB input 1: Counter count up enabled on the falling edge of GTETRGRB input	R/W
4	USGTRGCR	GTETRGRG Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGRG input 1: Counter count up enabled on the rising edge of GTETRGRG input	R/W
5	USGTRGCF	GTETRGRG Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGRG input 1: Counter count up enabled on the falling edge of GTETRGRG input	R/W
6	USGTRGDR	GTETRGRD Pin Rising Input Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTETRGRD input 1: Counter count up enabled on the rising edge of GTETRGRD input	R/W
7	USGTRGDF	GTETRGRD Pin Falling Input Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTETRGRD input 1: Counter count up enabled on the falling edge of GTETRGRD input	R/W
8	USCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count up enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	USCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count up enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	USCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count up enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	USCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count up enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	USCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count up enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	USCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Up Enable 0: Counter count up disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count up enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	USCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count up enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	USCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Up Enable 0: Counter count up disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count up enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W

Bit	Symbol	Function	R/W
16	USELCA	ELC_GPTA Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTA input 1: Counter count up enabled at the ELC_GPTA input	R/W
17	USELCB	ELC_GPTB Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTB input 1: Counter count up enabled at the ELC_GPTB input	R/W
18	USELCC	ELC_GPTC Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTC input 1: Counter count up enabled at the ELC_GPTC input	R/W
19	USELCD	ELC_GPTD Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTD input 1: Counter count up enabled at the ELC_GPTD input	R/W
20	USELCE	ELC_GPTE Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTE input 1: Counter count up enabled at the ELC_GPTE input	R/W
21	USELCF	ELC_GPTF Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTF input 1: Counter count up enabled at the ELC_GPTF input	R/W
22	USELCG	ELC_GPTG Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTG input 1: Counter count up enabled at the ELC_GPTG input	R/W
23	USELCH	ELC_GPTH Event Source Counter Count Up Enable 0: Counter count up disabled at the ELC_GPTH input 1: Counter count up enabled at the ELC_GPTH input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 5, 10 to 13

The GTUPSR sets the source to count up the GTCNT counter.

When at least one bit in the GTUPSR register is set to 1, the GTCNT counter is counted up by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

Number of increment in counting is one even when multiple sources are generated simultaneously.

Inputs from GTETRGN (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

#### **USGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Up Enable)**

The USGTRGAR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGA pin input.

#### **USGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Up Enable)**

The USGTRGAF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGA pin input.

#### **USGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Up Enable)**

The USGTRGBR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGB pin input.

#### **USGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Up Enable)**

The USGTRGBF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGB pin input.

#### **USGTRGCR bit (GTETRGC Pin Rising Input Source Counter Count Up Enable)**

The USGTRGCR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGC pin input.

#### **USGTRGCF bit (GTETRGC Pin Falling Input Source Counter Count Up Enable)**

The USGTRGCF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGC pin input.

#### **USGTRGDR bit (GTETRGD Pin Rising Input Source Counter Count Up Enable)**

The USGTRGDR bit enables or disables the GTCNT counter count up on the rising edge of the GTETRGD pin input.

**USGTRGDF bit (GTETRGD Pin Falling Input Source Counter Count Up Enable)**

The USGTRGDF bit enables or disables the GTCNT counter count up on the falling edge of the GTETRGD pin input.

**USCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Up Enable)**

The USCARBL bit enables or disables GTCNT counter count up on the rising edge of GTIOCnA pin input, when GTIOCnB input is 0.

**USCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Up Enable)**

The USCARBH bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

**USCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Up Enable)**

The USCAFBL bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

**USCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Up Enable)**

The USCAFBH bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

**USCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Up Enable)**

The USCBRAL bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

**USCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Up Enable)**

The USCBRAH bit enables or disables the GTCNT counter count up on the rising edge of the GTIOCnB pin input, when the GTIOCnA input is 1.

**USCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Up Enable)**

The USCBFAL bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnB pin input, when the GTIOCnA input is 0.

**USCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Up Enable)**

The USCBFAH bit enables or disables the GTCNT counter count up on the falling edge of the GTIOCnB pin input, when the GTIOCnA input is 1.

**USELCm bit (ELC\_GPTm Event Source Counter Count Up Enable) (m = A to H)**

The USELCm bit enables or disables the GTCNT counter count up at the ELC\_GPTm event input.

**21.2.9 GTDNSR : General PWM Timer Down Count Source Select Register**

Base address:  $GPT32n = 0x4032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT32n\_NS = 0x5032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT16m = 0x4032\_2000 + 0x0100 \times m$  (m = 10 to 13)  
 $GPT16m\_NS = 0x5032\_2000 + 0x0100 \times m$  (m = 10 to 13)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DSEL CH	DSEL CG	DSEL CF	DSEL CE	DSEL CD	DSEL CC	DSEL CB	DSEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DSCB FAH	DSCB FAL	DSCB RAH	DSCB RAL	DSCA FBH	DSCA FBL	DSCA RBH	DSCA RBL	DSGT RGDF	DSGT RGDR	DSGT RGCF	DSGT RGCR	DSGT RGBF	DSGT RBR	DSGT RGAF	DSGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DSGTRGAR	GTETRGA Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGA input 1: Counter count down enabled on the rising edge of GTETRGA input	R/W
1	DSGTRGAF	GTETRGA Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGA input 1: Counter count down enabled on the falling edge of GTETRGA input	R/W
2	DSGTRGBR	GTETRGB Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGB input 1: Counter count down enabled on the rising edge of GTETRGB input	R/W
3	DSGTRGBF	GTETRGB Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGB input 1: Counter count down enabled on the falling edge of GTETRGB input	R/W
4	DSGTRGCR	GTETRGC Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGC input 1: Counter count down enabled on the rising edge of GTETRGC input	R/W
5	DSGTRGCF	GTETRGC Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGC input 1: Counter count down enabled on the falling edge of GTETRGC input	R/W
6	DSGTRGDR	GTETRGD Pin Rising Input Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTETRGD input 1: Counter count down enabled on the rising edge of GTETRGD input	R/W
7	DSGTRGDF	GTETRGD Pin Falling Input Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTETRGD input 1: Counter count down enabled on the falling edge of GTETRGD input	R/W
8	DSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count down enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	DSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count down enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	DSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: Counter count down enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	DSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: Counter count down enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	DSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count down enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W



Bit	Symbol	Function	R/W
13	DSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Down Enable 0: Counter count down disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count down enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	DSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: Counter count down enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	DSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Down Enable 0: Counter count down disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: Counter count down enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	DSELCA	ELC_GPTA Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTA input 1: Counter count down enabled at the ELC_GPTA input	R/W
17	DSELCB	ELC_GPTB Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTB input 1: Counter count down enabled at the ELC_GPTB input	R/W
18	DSELCC	ELC_GPTC Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTC input 1: Counter count down enabled at the ELC_GPTC input	R/W
19	DSELCD	ELC_GPTD Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTD input 1: Counter count down enabled at the ELC_GPTD input	R/W
20	DSELCE	ELC_GPTE Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTE input 1: Counter count down enabled at the ELC_GPTE input	R/W
21	DSELCF	ELC_GPTF Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTF input 1: Counter count down enabled at the ELC_GPTF input	R/W
22	DSELCG	ELC_GPTG Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTG input 1: Counter count down enabled at the ELC_GPTG input	R/W
23	DSELCH	ELC_GPTF Event Source Counter Count Down Enable 0: Counter count down disabled at the ELC_GPTF input 1: Counter count down enabled at the ELC_GPTF input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 5, 10 to 13

The GTDNSR sets the source to count down the GTCNT counter.

When at least one bit in the GTDNSR register is set to 1, the GTCNT counter is counted down by the source that is set to 1 in this register. In this case, GTCR.TPCS has no effect.

Number of decrement in counting is one even when multiple sources are generated simultaneously.

Inputs from GTETRn (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

#### **DSGTRGAR bit (GTETRGA Pin Rising Input Source Counter Count Down Enable)**

The DSGTRGAR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGA pin input.

#### **DSGTRGAF bit (GTETRGA Pin Falling Input Source Counter Count Down Enable)**

The DSGTRGAF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGA pin input.

**DSGTRGBR bit (GTETRGB Pin Rising Input Source Counter Count Down Enable)**

The DSGTRGBR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGB pin input.

**DSGTRGBF bit (GTETRGB Pin Falling Input Source Counter Count Down Enable)**

The DSGTRGBF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGB pin input.

**DSGTRGCR bit (GTETRGC Pin Rising Input Source Counter Count Down Enable)**

The DSGTRGCR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGC pin input.

**DSGTRGCF bit (GTETRGC Pin Falling Input Source Counter Count Down Enable)**

The DSGTRGCF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGC pin input.

**DSGTRGDR bit (GTETRGD Pin Rising Input Source Counter Count Down Enable)**

The DSGTRGDR bit enables or disables the GTCNT counter count down on the rising edge of the GTETRGD pin input.

**DSGTRGDF bit (GTETRGD Pin Falling Input Source Counter Count Down Enable)**

The DSGTRGDF bit enables or disables the GTCNT counter count down on the falling edge of the GTETRGD pin input.

**DSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source Counter Count Down Enable)**

The DSCARBL bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnA pin input, when the GTIOCnB input is 0.

**DSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source Counter Count Down Enable)**

The DSCARBH bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

**DSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source Counter Count Down Enable)**

The DSCAFBL bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

**DSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source Counter Count Down Enable)**

The DSCAFBH bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

**DSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source Counter Count Down Enable)**

The DSCBRAL bit enables or disables the GTCNT counter count down on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

**DSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source Counter Count Down Enable)**

The DSCBRAH bit enables or disables the GTCNT counter count down on the rising edge of GTIOCnB pin input, when GTIOCnA input is 1.

**DSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source Counter Count Down Enable)**

The DSCBFAL bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

**DSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source Counter Count Down Enable)**

The DSCBFAH bit enables or disables the GTCNT counter count down on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

**DSELCm bit (ELC\_GPTm Event Source Counter Count Down Enable) (m = A to H)**

The DSELCm bit enables or disables the GTCNT counter count down at the ELC\_GPTm event input.

**21.2.10 GTICASR : General PWM Timer Input Capture Source Select Register A**

Base address:  $GPT32n = 0x4032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT32n\_NS = 0x5032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT16m = 0x4032\_2000 + 0x0100 \times m$  (m = 10 to 13)  
 $GPT16m\_NS = 0x5032\_2000 + 0x0100 \times m$  (m = 10 to 13)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	ASEL CH	ASEL CG	ASEL CF	ASEL CE	ASEL CD	ASEL CC	ASEL CB	ASEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASCB FAH	ASCB FAL	ASCB RAH	ASCB RAL	ASCA FBH	ASCA FBL	ASCA RBH	ASCA RBL	ASGT RGDF	ASGT RGDR	ASGT RGCF	ASGT RGCR	ASGT RGBF	ASGT RGBR	ASGT RGAF	ASGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ASGTRGAR	GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGA input 1: GTCCRA input capture enabled on the rising edge of GTETRGA input	R/W
1	ASGTRGAF	GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGA input 1: GTCCRA input capture enabled on the falling edge of GTETRGA input	R/W
2	ASGTRGBR	GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGB input 1: GTCCRA input capture enabled on the rising edge of GTETRGB input	R/W
3	ASGTRGBF	GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGB input 1: GTCCRA input capture enabled on the falling edge of GTETRGB input	R/W
4	ASGTRGCR	GTETRGC Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGC input 1: GTCCRA input capture enabled on the rising edge of GTETRGC input	R/W
5	ASGTRGCF	GTETRGC Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGC input 1: GTCCRA input capture enabled on the falling edge of GTETRGC input	R/W
6	ASGTRGDR	GTETRGD Pin Rising Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTETRGD input 1: GTCCRA input capture enabled on the rising edge of GTETRGD input	R/W
7	ASGTRGDF	GTETRGD Pin Falling Input Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTETRGD input 1: GTCCRA input capture enabled on the falling edge of GTETRGD input	R/W
8	ASCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W

Bit	Symbol	Function	R/W
9	ASCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	ASCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	ASCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRA input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	ASCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRA input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	ASCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRA input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	ASCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRA input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W
15	ASCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRA input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	ASELCA	ELC_GPTA Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTA input 1: GTCCRA input capture enabled at the ELC_GPTA input	R/W
17	ASELCB	ELC_GPTB Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTB input 1: GTCCRA input capture enabled at the ELC_GPTB input	R/W
18	ASELCC	ELC_GPTC Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTC input 1: GTCCRA input capture enabled at the ELC_GPTC input	R/W
19	ASELCD	ELC_GPTD Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTD input 1: GTCCRA input capture enabled at the ELC_GPTD input	R/W
20	ASELCE	ELC_GPTE Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTE input 1: GTCCRA input capture enabled at the ELC_GPTE input	R/W
21	ASELCF	ELC_GPTF Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTF input 1: GTCCRA input capture enabled at the ELC_GPTF input	R/W

Bit	Symbol	Function	R/W
22	ASELCG	ELC_GPTG Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTG input 1: GTCCRA input capture enabled at the ELC_GPTG input	R/W
23	ASELCH	ELC_GPTH Event Source GTCCRA Input Capture Enable 0: GTCCRA input capture disabled at the ELC_GPTH input 1: GTCCRA input capture enabled at the ELC_GPTH input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 5, 10 to 13

The GTICASR sets the source of input capture for GTCCRA.

When at least one bit among bits in the GTICASR register is set to 1, input capture operation making the GTCCRA register as an input capture register is performed.

Inputs from GTETRGN (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

#### **ASGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRA Input Capture Enable)**

The ASGTRGAR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGA pin input.

#### **ASGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRA Input Capture Enable)**

The ASGTRGAF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGA pin input.

#### **ASGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRA Input Capture Enable)**

The ASGTRGBR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGB pin input.

#### **ASGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRA Input Capture Enable)**

The ASGTRGBF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGB pin input.

#### **ASGTRGCR bit (GTETRGC Pin Rising Input Source GTCCRA Input Capture Enable)**

The ASGTRGCR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGC pin input.

#### **ASGTRGCF bit (GTETRGC Pin Falling Input Source GTCCRA Input Capture Enable)**

The ASGTRGCF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGC pin input.

#### **ASGTRGDR bit (GTETRGD Pin Rising Input Source GTCCRA Input Capture Enable)**

The ASGTRGDR bit enables or disables the input capture for GTCCRA on the rising edge of the GTETRGD pin input.

#### **ASGTRGDF bit (GTETRGD Pin Falling Input Source GTCCRA Input Capture Enable)**

The ASGTRGDF bit enables or disables the input capture for GTCCRA on the falling edge of the GTETRGD pin input.

#### **ASCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable)**

The ASCARBL bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 0.

#### **ASCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRA Input Capture Enable)**

The ASCARBH bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

#### **ASCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRA Input Capture Enable)**

The ASCAFBL bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

**ASCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRA Input Capture Enable)**

The ASCAFBH bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnA pin input, when the GTIOCnB input is 1.

**ASCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable)**

The ASCBRAL bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnB pin input, when the GTIOCnA input is 0.

**ASCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRA Input Capture Enable)**

The ASCBRAH bit enables or disables the input capture for GTCCRA on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

**ASCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRA Input Capture Enable)**

The ASCBFAL bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

**ASCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRA Input Capture Enable)**

The ASCBFAH bit enables or disables the input capture for GTCCRA on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

**ASELCm bit (ELC\_GPTm Event Source Counter GTCCRA Input Capture Enable) (m = A to H)**

The ASELCm bit enables or disables the input capture for GTCCRA at the ELC\_GPTm event input.

**21.2.11 GTICBSR : General PWM Timer Input Capture Source Select Register B**

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	BSEL CH	BSEL CG	BSEL CF	BSEL CE	BSEL CD	BSEL CC	BSEL CB	BSEL CA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSCB FAH	BSCB FAL	BSCB RAH	BSCB RAL	BSCA FBH	BSCA FBL	BSCA RBH	BSCA RBL	BSGT RGDF	BSGT RGDR	BSGT RGCF	BSGT RGCR	BSGT RGBF	BSGT RGBR	BSGT RGAF	BSGT RGAR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BSGTRGAR	GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGA input 1: GTCCRB input capture enabled on the rising edge of GTETRGA input	R/W
1	BSGTRGAF	GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGA input 1: GTCCRB input capture enabled on the falling edge of GTETRGA input	R/W
2	BSGTRGBR	GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGB input 1: GTCCRB input capture enabled on the rising edge of GTETRGB input	R/W

Bit	Symbol	Function	R/W
3	BSGTRGBF	GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGC input 1: GTCCRB input capture enabled on the falling edge of GTETRGC input	R/W
4	BSGTRGCR	GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGC input 1: GTCCRB input capture enabled on the rising edge of GTETRGC input	R/W
5	BSGTRGCF	GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGC input 1: GTCCRB input capture enabled on the falling edge of GTETRGC input	R/W
6	BSGTRGDR	GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTETRGC input 1: GTCCRB input capture enabled on the rising edge of GTETRGC input	R/W
7	BSGTRGDF	GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTETRGC input 1: GTCCRB input capture enabled on the falling edge of GTETRGC input	R/W
8	BSCARBL	GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 0	R/W
9	BSCARBH	GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCnA input when GTIOCnB input is 1	R/W
10	BSCAFBL	GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 0	R/W
11	BSCAFBH	GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnA input when GTIOCnB input is 1 1: GTCCRB input capture enabled on the falling edge of GTIOCnA input when GTIOCnB input is 1	R/W
12	BSCBRAL	GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRB input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 0	R/W
13	BSCBRAH	GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the rising edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRB input capture enabled on the rising edge of GTIOCnB input when GTIOCnA input is 1	R/W
14	BSCBFAL	GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 0 1: GTCCRB input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 0	R/W

Bit	Symbol	Function	R/W
15	BSCBFAH	GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled on the falling edge of GTIOCnB input when GTIOCnA input is 1 1: GTCCRB input capture enabled on the falling edge of GTIOCnB input when GTIOCnA input is 1	R/W
16	BSELCA	ELC_GPTA Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTA input 1: GTCCRB input capture enabled at the ELC_GPTA input	R/W
17	BSELCB	ELC_GPTB Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTB input 1: GTCCRB input capture enabled at the ELC_GPTB input	R/W
18	BSELCC	ELC_GPTC Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTC input 1: GTCCRB input capture enabled at the ELC_GPTC input	R/W
19	BSELCD	ELC_GPTD Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTD input 1: GTCCRB input capture enabled at the ELC_GPTD input	R/W
20	BSELCE	ELC_GPTE Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTE input 1: GTCCRB input capture enabled at the ELC_GPTE input	R/W
21	BSELCF	ELC_GPTF Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTF input 1: GTCCRB input capture enabled at the ELC_GPTF input	R/W
22	BSELCG	ELC_GPTG Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTG input 1: GTCCRB input capture enabled at the ELC_GPTG input	R/W
23	BSELCH	ELC_GPTH Event Source GTCCRB Input Capture Enable 0: GTCCRB input capture disabled at the ELC_GPTH input 1: GTCCRB input capture enabled at the ELC_GPTH input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 5, 10 to 13

The GTICBSR sets the source of input capture for GTCCRB.

When at least one bit among bits in the GTICBSR register is set to 1, input capture operation making the GTCCRB register as an input capture register is performed.

Inputs from GTETRGN (n = A to D) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

#### **BSGTRGAR bit (GTETRGA Pin Rising Input Source GTCCRB Input Capture Enable)**

The BSGTRGAR bit enables or disables the input capture for GTCCRB on the rising edge of the GTETRGA pin input.

#### **BSGTRGAF bit (GTETRGA Pin Falling Input Source GTCCRB Input Capture Enable)**

The BSGTRGAF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGA pin input.

#### **BSGTRGBR bit (GTETRGB Pin Rising Input Source GTCCRB Input Capture Enable)**

The BSGTRGBR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGB pin input.

#### **BSGTRGBF bit (GTETRGB Pin Falling Input Source GTCCRB Input Capture Enable)**

The BSGTRGBF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGB pin input.

#### **BSGTRGCR bit (GTETRGC Pin Rising Input Source GTCCRB Input Capture Enable)**

The BSGTRGCR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGC pin input.

#### **BSGTRGCF bit (GTETRGC Pin Falling Input Source GTCCRB Input Capture Enable)**

The BSGTRGCF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGC pin input.



**BSGTRGDR bit (GTETRGD Pin Rising Input Source GTCCRB Input Capture Enable)**

The BSGTRGDR bit enables or disables the input capture for GTCCRB on the rising edge of GTETRGD pin input.

**BSGTRGDF bit (GTETRGD Pin Falling Input Source GTCCRB Input Capture Enable)**

The BSGTRGDF bit enables or disables the input capture for GTCCRB on the falling edge of the GTETRGD pin input.

**BSCARBL bit (GTIOCnA Pin Rising Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable)**

The BSCARBL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnA pin input, when the GTIOCnB input is 0.

**BSCARBH bit (GTIOCnA Pin Rising Input during GTIOCnB Value High Source GTCCRB Input Capture Enable)**

The BSCARBH bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnA pin input, when GTIOCnB input is 1.

**BSCAFBL bit (GTIOCnA Pin Falling Input during GTIOCnB Value Low Source GTCCRB Input Capture Enable)**

The BSCAFBL bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 0.

**BSCAFBH bit (GTIOCnA Pin Falling Input during GTIOCnB Value High Source GTCCRB Input Capture Enable)**

The BSCAFBH bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnA pin input, when GTIOCnB input is 1.

**BSCBRAL bit (GTIOCnB Pin Rising Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable)**

The BSCBRAL bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 0.

**BSCBRAH bit (GTIOCnB Pin Rising Input during GTIOCnA Value High Source GTCCRB Input Capture Enable)**

The BSCBRAH bit enables or disables the input capture for GTCCRB on the rising edge of the GTIOCnB pin input, when GTIOCnA input is 1.

**BSCBFAL bit (GTIOCnB Pin Falling Input during GTIOCnA Value Low Source GTCCRB Input Capture Enable)**

The BSCBFAL bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 0.

**BSCBFAH bit (GTIOCnB Pin Falling Input during GTIOCnA Value High Source GTCCRB Input Capture Enable)**

The BSCBFAH bit enables or disables the input capture for GTCCRB on the falling edge of the GTIOCnB pin input, when GTIOCnA input is 1.

**BSELCm bit (ELC\_GPTm Event Source Counter GTCCRB Input Capture Enable) (m = A to H)**

The BSELCm bit enables or disables the input capture for GTCCRB at the ELC\_GPTm event input.

**21.2.12 GTCR : General PWM Timer Control Register**

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16		
Bit field:	—	—	—	—	—	TPCS[3:0]						—	—	—	—	MD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CST		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

Bit	Symbol	Function	R/W
0	CST	Count Start 0: Count operation is stopped 1: Count operation is performed	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
18:16	MD[2:0]	Mode Select 0 0 0: Saw-wave PWM mode (single buffer or double buffer possible) 0 0 1: Saw-wave one-shot pulse mode (fixed buffer operation) 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer is possible) 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer is possible) 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 1 1 1: Setting prohibited	R/W
22:19	—	These bits are read as 0. The write value should be 0.	R/W
26:23	TPCS[3:0]	Timer Prescaler Select 0 0 0 0: PCLKD/1 0 0 0 1: PCLKD/2 0 0 1 0: PCLKD/4 0 0 1 1: PCLKD/8 0 1 0 0: PCLKD/16 0 1 0 1: PCLKD/32 0 1 1 0: PCLKD/64 0 1 1 1: Setting prohibited 1 0 0 0: PCLKD/256 1 0 0 1: Setting prohibited 1 0 1 0: PCLKD/1024 1 0 1 1: Setting prohibited 1 1 0 0: GTETRGA (Via the POEG) 1 1 0 1: GTETRGB (Via the POEG) 1 1 1 0: GTETRGC (Via the POEG) 1 1 1 1: GTETRGD (Via the POEG)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

The GTCR controls GTCNT.

### CST bit (Count Start)

The CST bit controls the GTCNT counter start and stop.

[Setting conditions]

- The GTSTR value where the channel number associated with the bit number is set to 1 with the GTSSR.CSTRT bit at 1
- The ELC event input, the external trigger, or the GTIOCN<sub>A</sub>/GTIOCN<sub>B</sub> input that are enabled by GTSSR for the starting counter source, occurs (n = 0 to 5, 10 to 13)
- 1 is written by software directly.

[Clearing conditions]

- The GTSTP value where the channel number associated with the bit number is set to 1 with the GTPSR.CSTOP bit at 1

- The ELC event input, the external trigger, or the GTIOCnA/GTIOCnB input enabled by GTPSR as the counter stop source, occurs (n = 0 to 5, 10 to 13)
- 0 is written by software directly.
- When the period count function is finished while the GTPC.ASTP bit is 1.

**MD[2:0] bits (Mode Select)**

The MD[2:0] bits select the GPT operating mode.

Only the MD[2] bit is valid at input capture. Counting in saw-wave mode is performed with 0 for the MD[2] bit, and counting in triangle-wave mode is performed with 1 for the MD[2] bit. The MD bits must be set while the GTCNT operation is stopped. During the event count operation (when at least one bit among the bits for the GTUPSR and GTDNSR registers is set to 1), setting of the MD bits are ignored, where counting in saw-wave or triangle-wave modes is not performed. Instead, up-counting or down-counting by a source set by the GTUPSR and GTDNSR registers is performed.

**TPCS[3:0] bits (Timer Prescaler Select)**

The TPCS[3:0] bits select the clock for GTCNT. A clock prescaler can be selected independently for each channel. The TPCS[3:0] bits must be set while the GTCNT operation is stopped. When the GTETRGA, GTETRGB, GTETRCG, or GTETRGD is selected, output for the POEG at rising becomes a clock source. Set the polarity of these signals with the POEG.

**21.2.13 GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register**

Base address:  $GPT32n = 0x4032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT32n\_NS = 0x5032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT16m = 0x4032\_2000 + 0x0100 \times m$  (m = 10 to 13)  
 $GPT16m\_NS = 0x5032\_2000 + 0x0100 \times m$  (m = 10 to 13)

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	OBDT YR	OBDT YF	OBDTY[1:0]	—	—	—	—	OADT YR	OADT YF	OADTY[1:0]	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UDF	UD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	UD	Count Direction Setting 0: GTCNT counts down 1: GTCNT counts up	R/W
1	UDF	Forcible Count Direction Setting 0: Not forcibly set 1: Forcibly set	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
17:16	OADTY[1:0]	GTIOCnA Output Duty Setting 0 0: GTIOCnA pin duty depends on the compare match 0 1: GTIOCnA pin duty depends on the compare match 1 0: GTIOCnA pin duty 0% 1 1: GTIOCnA pin duty 100%	R/W
18	OADTYF	Forcible GTIOCnA Output Duty Setting 0: Not forcibly set 1: Forcibly set	R/W

Bit	Symbol	Function	R/W
19	OADTYR	GTIOcNA Output Value Selecting after Releasing 0%/100% Duty Setting 0: The function selected by the GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOA[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	OBDTY[1:0]	GTIOcNB Output Duty Setting 0 0: GTIOcNB pin duty depends on the compare match 0 1: GTIOcNB pin duty depends on the compare match 1 0: GTIOcNB pin duty 0% 1 1: GTIOcNB pin duty 100%	R/W
26	OBDTYF	Forcible GTIOcNB Output Duty Setting 0: Not forcibly set 1: Forcibly set	R/W
27	OBDTYR	GTIOcNB Output Value Selecting after Releasing 0%/100% Duty Setting 0: The function selected by the GTIOB[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting. 1: The function selected by the GTIOB[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 5, 10 to 13

The GTUDDTYC sets the direction in which the GTCNT counts (up-counting or down-counting), and sets the duty of the GTIOcNA/GTIOcNB pin output.

The setting is invalid during the event count operation.

Count Direction:

- In saw-wave mode.  
When the UD value is set to 0 during up-counting, the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes the GTPR value). When the UD value is set to 1 during down-counting, the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).  
When the UD value changes from 1 to 0 with the UDF bit being 0 and while counting stops, the counter starts up-counting and the count direction changes at an overflow (the timing synchronous with count clock after the GTCNT value becomes the GTPR value). When the UD value changes from 0 to 1 with the UDF bit being 0 and while counting stops, the counter starts down-counting and the count direction changes at an underflow (the timing synchronous with count clock after the GTCNT value becomes 0).  
When the UDF bit is set to 1 while counting stops, the UD bit value is reflected in the count direction when counting starts.
- In triangle-wave mode.  
When the UD value changes during counting, the count direction does not change. When the UD value changes while the UDF bit is 0 and counting stops, the change is not reflected in the count direction when counting starts.  
When the UDF bit is set to 1 while counting is stopped, the UD value is reflected in the count direction when counting starts.

### UD bit (Count Direction Setting)

The UD bit sets the count direction (up-counting or down-counting) for GTCNT.

### UDF bit (Forcible Count Direction Setting)

The UDF bit forcibly sets the count direction when GTCNT starts operation as the UD value. Only 0 should be written to this bit during counter operation. When 1 is written to this bit while counting stops, return this bit to 0 before counting starts.

Output duty

- In saw-wave mode.

When the OADTY/OBDTY value changes during up-counting, the duty is reflected at an overflow (GTCNT = GTPR). When the OADTY/OBDTY value is changed during down-counting, the duty is reflected at an underflow (GTCNT = 0).

When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 0 and while counting stops the output duty is not reflected at the starting counter operation. When the count direction is up, the output duty is reflected at an overflow (GTCNT = GTPR). When the count direction is down, the output duty is reflected at an underflow (GTCNT = 0).

When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at starting counter operation.

- In triangle-wave mode.

When the OADTY/OBDTY value changes during counting, the duty is reflected at an underflow.

When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 0 and while counting stops, the output duty is not reflected at the starting counter operation. The output duty is reflected at an underflow.

When the OADTY/OBDTY value is changed with the OADTYF/OBDTYF bit being 1 and while counting stops, the output duty is reflected at starting counter operation.

In both saw-wave mode and triangle-wave mode, when the OADTYF/OBDTYF bit is set back to 0 and the OADTY[1:0]/OBDTY[1:0] bits are set after setting the OADTYF/OBDTYF bit to 1 and setting the OADTY[1:0]/OBDTY[1:0] bits for the duty of first cycle while count operation is stopped, these duty-cycle set during stopping count operation are reflected in the first cycle and the second cycle after starting count operation.

### OmdTY[1:0] bits (GTIOCnm Output Duty Setting) (m = A, B)

The OmdTY[1:0] bits set the output duty (0%, 100% or compare match control) of the GTIOCnm pin.

### OmdTYF bit (Forcible GTIOCnm Output Duty Setting) (m = A, B)

The OmdTYF bit forcibly sets the output duty cycle to the OmdTY setting. Set this bit to 0 during counter operation.

### OmdTYR bit (GTIOCnm Output Value Selecting after Releasing 0%/100% Duty Setting) (m = A, B)

The OmdTYR bit selects the value that is the object of output retained or toggled at cycle end, when the control changes from 0% or 100% duty setting to compare match for the GTIOCnm pin and GTIOR.GTIOm[3:2] bits are set to 00b (output retained at cycle end) or the GTIOR.GTIOm[3:2] bits are set to 11b (output toggled at cycle end).

The GPT internally continues to perform compare match operation during duty-cycle 0% or 100% operation. When the OmdTYR bit is 1, the value after the period has elapsed due this compare match operation is target for the GTIOm[3:2] bits.

## 21.2.14 GTIOR : General PWM Timer I/O Control Register

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCSB[1:0]		NFBEN	—	—	OBDF[1:0]		OBE	OBHLD	OBDFLT	—	GTIOB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NFCSA[1:0]		NFAEN	—	—	OADF[1:0]		OAE	OAHL D	OADFLT	—	GTIOA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	GTIOA[4:0]	GTIOCnA Pin Function Select See <a href="#">Table 21.4</a> .	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	OADFLT	GTIOCnA Pin Output Value Setting at the Count Stop 0: The GTIOCnA pin outputs low when counting stops 1: The GTIOCnA pin outputs high when counting stops	R/W
7	OAHL D	GTIOCnA Pin Output Setting at the Start/Stop Count 0: The GTIOCnA pin output level at the start or stop of counting depends on the register setting 1: The GTIOCnA pin output level is retained at the start or stop of counting	R/W
8	OAE	GTIOCnA Pin Output Enable 0: Output is disabled 1: Output is enabled	R/W
10:9	OADF[1:0]	GTIOCnA Pin Disable Value Setting 0 0: None of the below options are specified 0 1: GTIOCnA pin is set to Hi-Z in response to controlling the output negation 1 0: GTIOCnA pin is set to 0 in response to controlling the output negation 1 1: GTIOCnA pin is set to 1 in response to controlling the output negation	R/W
12:11	—	These bits are read as 0. The write value should be 0.	R/W
13	NFAEN	Noise Filter A Enable 0: The noise filter for the GTIOCnA pin is disabled 1: The noise filter for the GTIOCnA pin is enabled	R/W
15:14	NFCSA[1:0]	Noise Filter A Sampling Clock Select 0 0: PCLKD/1 0 1: PCLKD/4 1 0: PCLKD/16 1 1: PCLKD/64	R/W
20:16	GTIOB[4:0]	GTIOCnB Pin Function Select See <a href="#">Table 21.4</a> .	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	OBDFLT	GTIOCnB Pin Output Value Setting at the Count Stop 0: The GTIOCnB pin outputs low when counting stops 1: The GTIOCnB pin outputs high when counting stops	R/W
23	OBHL D	GTIOCnB Pin Output Setting at the Start/Stop Count 0: The GTIOCnB pin output level at the start/stop of counting depends on the register setting 1: The GTIOCnB pin output level is retained at the start/stop of counting	R/W
24	OBE	GTIOCnB Pin Output Enable 0: Output is disabled 1: Output is enabled	R/W
26:25	OBDF[1:0]	GTIOCnB Pin Disable Value Setting 0 0: None of the below options are specified 0 1: GTIOCnB pin is set to Hi-Z in response to controlling the output negation 1 0: GTIOCnB pin is set to 0 in response to controlling the output negation 1 1: GTIOCnB pin is set to 1 in response to controlling the output negation	R/W
28:27	—	These bits are read as 0. The write value should be 0.	R/W
29	NFBEN	Noise Filter B Enable 0: The noise filter for the GTIOCnB pin is disabled 1: The noise filter for the GTIOCnB pin is enabled	R/W
31:30	NFCSB[1:0]	Noise Filter B Sampling Clock Select 0 0: PCLKD/1 0 1: PCLKD/4 1 0: PCLKD/16 1 1: PCLKD/64	R/W

Note: n = 0 to 5, 10 to 13

The GTIOR sets the functions of the GTIOCnA and GTIOCnB pins. (n = 0 to 5, 10 to 13)

#### GTIOA[4:0] bits (GTIOCnA Pin Function Select)

The GTIOA[4:0] bits select the GTIOCnA pin function. For details, see [Table 21.4](#).

**OADFLT bit (GTIOCnA Pin Output Value Setting at the Count Stop)**

The OADFLT bit sets whether the GTIOCnA pin outputs high or low when counting stops.

**OAHLD bit (GTIOCnA Pin Output Setting at the Start/Stop Count)**

The OAHLD bit specifies whether the GTIOCnA pin output level is retained or the level at the start or stop of counting depends on the register setting.

When the OAHLD bit is set to 0:

- The value specified in bit [4] of the GTIOA[4:0] bits is output when counting starts
- The value specified in the OADFLT bit is output when counting stops
- If the OADFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OAHLD bit is set to 1:

- The output is retained when counting starts or stops.

**OAE bit (GTIOCnA Pin Output Enable)**

The OAE bit disables or enables the GTIOCnA pin output.

When GTCCRA register is used as the input capture register (at least one bit in the GTICASR register is set to 1), the GTIOCnA pin does not output regardless of the OAE bit value.

**OADF[1:0] bits (GTIOCnA Pin Disable Value Setting)**

The OADF[1:0] bits select the output value of the GTIOCnA pin in response to a request to disable output from the POEG.

**NFAEN bit (Noise Filter A Enable)**

The NFAEN bit disables or enables the noise filter for input from the GTIOCnA pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

**NFCSA[1:0] bits (Noise Filter A Sampling Clock Select)**

The NFCSA[1:0] bits set the sampling interval for the noise filter of the GTIOCnA pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

**GTIOB[4:0] bits (GTIOCnB Pin Function Select)**

The GTIOB[4:0] bits select the GTIOCnB pin function. For details, see [Table 21.4](#).

**OBDFLT bit (GTIOCnB Pin Output Value Setting at the Count Stop)**

The OBDFLT bit sets whether the GTIOCnB pin outputs high or low when counting stops.

**OBHLD bit (GTIOCnB Pin Output Setting at the Start/Stop Count)**

The OBHLD bit specifies whether the GTIOCnB pin output level is retained or the level at the start or stop of counting depends on the register setting.

When the OBHLD bit is set to 0:

- The value specified in bit [4] of the GTIOB[4:0] bits is output when counting starts
- The value specified in the OBDFLT bit is output when counting stops
- If the OBDFLT bit is modified while counting stops, the new value is immediately reflected in the output.

When the OBHLD bit is set to 1:

- The output is retained when counting starts or stops.

**OBE bit (GTIOCnB Pin Output Enable)**

The OBE bit disables or enables the GTIOCnB pin output.

When GTCCRB register is used as the input capture register (at least one bit in the GTICBSR register is set to 1), the GTIOCnB pin does not output regardless of the OBE bit value.

**OBDF[1:0] bits (GTIOCnB Pin Disable Value Setting)**

The OBDF[1:0] bits select the output value of the GTIOCnB pin in response to a request to disable output from the POEG.

**NFBEN bit (Noise Filter B Enable)**

The NFBEN bit disables or enables the noise filter for input from the GTIOCnB pin. Because changing the value of the bit might lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the GTIOR register before doing so.

**NFCBSB[1:0] bits (Noise Filter B Sampling Clock Select)**

The NFCBSB[1:0] bits set the sampling interval for the noise filter of the GTIOCnB pin. When setting these bits, wait for 2 cycles of the selected sampling interval before setting the input capture function.

**Table 21.4 Settings of GTIOA[4:0] and GTIOB[4:0] bits (1 of 2)**

GTIOA/GTIOB[4:0] bits					Function		
b4	b3	b2	b1	b0	b4	b3, b2 <sup>*1</sup> *2 *3	b1, b0 <sup>*2</sup>
0	0	0	0	0	Initial output is low	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	0	1	0			High output at GTCCRA/GTCCRB compare match
0	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	1	1	0			High output at GTCCRA/GTCCRB compare match
0	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	0	1	0			High output at GTCCRA/GTCCRB compare match
0	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
0	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	1	1	0			High output at GTCCRA/GTCCRB compare match
0	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match



**Table 21.4 Settings of GTIOA[4:0] and GTIOB[4:0] bits (2 of 2)**

GTIOA/GTIOB[4:0] bits					Function		
b4	b3	b2	b1	b0	b4	b3, b2*1 *2 *3	b1, b0*2
1	0	0	0	0	Initial output is high	Output retained at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	0	1	0			High output at GTCCRA/GTCCRB compare match
1	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	1	0	0		Low output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	1	1	0			High output at GTCCRA/GTCCRB compare match
1	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	0	0	0		High output at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	0	1	0			High output at GTCCRA/GTCCRB compare match
1	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	1	0	0		Output toggled at cycle end	Output retained at GTCCRA/GTCCRB compare match
1	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	1	1	0			High output at GTCCRA/GTCCRB compare match
1	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

- Note 1. The cycle end means an overflow (GTCNT changes from GTPR to 0 in up-counting), an underflow (GTCNT changes from 0 to GTPR in down-counting), or counter clearing for saw-wave mode, and means a trough (GTCNT changes from 0 to 1) for triangle-wave mode.
- Note 2. When the timing of a cycle end and the timing of a GTCCRA/GTCCRB compare match are the same in a compare-match operation, the b3 and b2 settings are given priority in saw-wave PWM mode, and the b1 and b0 settings are given priority in any other mode.
- Note 3. In event count operation where at least one bit in GTUPSR or GTDNSR is set to 1, the setting of b3 and b2 is ignored.

### 21.2.15 GTINTAD : General PWM Timer Interrupt Output Setting Register

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	GRPA BL	GRPA BH	—	—	—	GRP[1:0]	—	—	—	—	—	ADTR BDEN	ADTR BUEN	ADTR ADEN	ADTR AUEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	ADTRAUEN	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W

Bit	Symbol	Function	R/W
17	ADTRADEN	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
18	ADTRBUEN	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
19	ADTRBDEN	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable 0: A/D conversion start request is disabled. 1: A/D conversion start request is enabled.	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GRP[1:0]	Output Disable Source Select 0 0: Group A output disable source is selected 0 1: Group B output disable source is selected 1 0: Group C output disable source is selected 1 1: Group D output disable source is selected	R/W
28:26	—	These bits are read as 0. The write value should be 0.	R/W
29	GRPABH	Same Time Output Level High Disable Request Enable 0: Same time output level high disable request disabled 1: Same time output level high disable request enabled	R/W
30	GRPABL	Same Time Output Level Low Disable Request Enable 0: Same time output level low disable request disabled 1: Same time output level low disable request enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The GTINTAD enables or disables interrupt requests, A/D conversion start request, and output disable requests.

#### **ADTRAUEN bit (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)**

This bit enables or disables A/D conversion start requests generated by GTADTRA register compare matches during GTCNT counter up-counting.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

#### **ADTRADEN bit (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)**

This bit enables or disables A/D conversion start requests generated by GTADTRA register compare matches during GTCNT counter down-counting.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

#### **ADTRBUEN bit (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Enable)**

This bit enables or disables A/D conversion start requests generated by GTADTRB register compare matches during GTCNT counter up-counting.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

#### **ADTRBDEN bit (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Enable)**

This bit enables or disables A/D conversion start requests generated by GTADTRB register compare matches during GTCNT counter down-counting.

The setting is invalid during the event count operation, and A/D conversion start request is not generated.

**GRP[1:0] bits (Output Disable Source Select)**

These bits select the group of output disable request from GPT to POEG and the group of output disable for GTIOCnA pin and GTIOCnB pin from POEG to GPT.

The output disable request to POEG is output to the group selected in the GRP[1:0] bit, with dead-time errors, simultaneous high output, and simultaneous low output factors following their respective disable request enable bits.

GTST.ODF shows the request of the output disable source group that is selected with the GRP[1:0] bits. Set the GRP[1:0] bits when both GTIOR.OAE and GTIOR.OBE bits are 0.

If the POEG other than the group connected to GPT is selected in the GRP[1:0] bits, ODF flag is always 0, and the status never change to the output disable state.

**GRPABH bit (Same Time Output Level High Disable Request Enable)**

The GRPABH bit enables or disables the output disable request when the GTIOCnA pin and GTIOCnB pin output 1 at the same time.

**GRPABL bit (Same Time Output Level Low Disable Request Enable)**

The GRPABL bit enables or disables the output disable request when the GTIOCnA pin and GTIOCnB pin output 0 at the same time.

**21.2.16 GTST : General PWM Timer Status Register**

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0x3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PCF	OABL F	OABH F	—	—	—	—	ODF	—	—	—	—	ADTR BDF	ADTR BUF	ADTR ADF	ADTR AUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TUCF	—	—	—	—	—	—	—	TCFP U	TCFP O	TCFF	TCFE	TCFD	TCFC	TCFB	TCFA
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCFA	Input Capture/Compare Match Flag A 0: No input capture/compare match of GTCCRA is generated 1: An input capture/compare match of GTCCRA is generated	R/W <sup>1</sup>
1	TCFB	Input Capture/Compare Match Flag B 0: No input capture/compare match of GTCCRB is generated 1: An input capture/compare match of GTCCRB is generated	R/W <sup>1</sup>
2	TCFC	Input Compare Match Flag C 0: No compare match of GTCCRC is generated 1: A compare match of GTCCRC is generated	R/W <sup>1</sup>
3	TCFD	Input Compare Match Flag D 0: No compare match of GTCCRD is generated 1: A compare match of GTCCRD is generated	R/W <sup>1</sup>
4	TCFE	Input Compare Match Flag E 0: No compare match of GTCCRE is generated 1: A compare match of GTCCRE is generated	R/W <sup>1</sup>
5	TCFF	Input Compare Match Flag F 0: No compare match of GTCCRF is generated 1: A compare match of GTCCRF is generated	R/W <sup>1</sup>

Bit	Symbol	Function	R/W
6	TCFPO	Overflow Flag 0: No overflow (crest) occurred 1: An overflow (crest) occurred	R/W <sup>*1</sup>
7	TCFPU	Underflow Flag 0: No underflow (trough) occurred 1: An underflow (trough) occurred	R/W <sup>*1</sup>
14:8	—	These bits are read as 0. The write value should be 0.	R/W
15	TUCF	Count Direction Flag 0: GTCNT counter counts downward 1: GTCNT counter counts upward	R
16	ADTRAUF	GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag 0: No GTADTRA register compare match has occurred in up-counting. 1: A GTADTRA register compare match has occurred in up-counting.	R/W <sup>*1</sup>
17	ADTRADF	GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag 0: No GTADTRA register compare match has occurred in down-counting. 1: A GTADTRA register compare match has occurred in down-counting.	R/W <sup>*1</sup>
18	ADTRBUF	GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag 0: No GTADTRB register compare match has occurred in up-counting. 1: A GTADTRB register compare match has occurred in up-counting.	R/W <sup>*1</sup>
19	ADTRBDF	GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag 0: No GTADTRB register compare match has occurred in down-counting. 1: A GTADTRB register compare match has occurred in down-counting.	R/W <sup>*1</sup>
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	ODF	Output Disable Flag 0: No output disable request is generated 1: An output disable request is generated	R
28:25	—	These bits are read as 0. The write value should be 0.	R/W
29	OABHF	Same Time Output Level High Flag 0: No simultaneous generation of 1 both for the GTIOCA and GTIOCB pins has occurred. 1: A simultaneous generation of 1 both for the GTIOCA and GTIOCB pins has occurred.	R
30	OABLF	Same Time Output Level Low Flag 0: No simultaneous generation of 0 both for the GTIOCA and GTIOCB pins has occurred. 1: A simultaneous generation of 0 both for the GTIOCA and GTIOCB pins has occurred.	R
31	PCF <sup>*2</sup>	Period Count Function Finish Flag 0: No period count function finish has occurred 1: A period count function finish has occurred	R/W <sup>*1</sup>

Note 1. Only 0 can be written to this bit. Do not write 1.

When clearing the ADTRAUF, ADTRADF, ADTRBUF, or ADTRBDF flag, be sure to write 0 only to the target flag or flags for clearing and to write 1 to the other flags not for clearing.

Note 2. This bit is only available in GPT320 to GPT323, GPT1610. In GPT324, GPT325, GPT1611 to GPT1613, this bit is read as 0. The write value should be 0.

The GTST indicates the status of the GPT.

### TCFA flag (Input Capture/Compare Match Flag A)

The TCFA flag indicates the status for the input capture or compare match of GTCCRA.

[Setting conditions]

- GTCNT = GTCCRA, when the GTCCRA register functions as a compare match register
- GTCNT counter value is transferred to GTCCRA by the input capture signal when the GTCCRA register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

### TCFB flag (Input Capture/Compare Match Flag B)

The TCFB flag indicates the status for the input capture or compare match of GTCCRB.

[Setting conditions]

- $GTCNT = GTCCRB$ , when the GTCCRB register functions as a compare match register
- GTCNT counter value is transferred to GTCCRB by the input capture signal when the GTCCRB register functions as an input capture register.

[Clearing condition]

- 0 is written to this flag.

### TCFC flag (Input Compare Match Flag C)

The TCFC flag indicates the status for the compare match of GTCCRC.

When GTCCRC performs buffer operation, GTCCRC doesn't perform compare match.

[Setting condition]

- $GTCNT = GTCCRC$ .

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- $GTCR.MD[2:0] = 001b$  (saw-wave one-shot pulse mode)
- $GTCR.MD[2:0] = 110b$  (triangle-wave PWM mode 3)
- $GTBER.CCRA[1:0] = 01b, 10b, 11b$  (GTCCRC performs buffer operation).

### TCFD flag (Input Compare Match Flag D)

The TCFD flag indicates the status for the compare match of GTCCRD.

When GTCCRD performs buffer operation, GTCCRD doesn't perform compare match.

[Setting condition]

- $GTCNT = GTCCRD$ .

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- $GTCR.MD[2:0] = 001b$  (saw-wave one-shot pulse mode)
- $GTCR.MD[2:0] = 110b$  (Triangle-wave PWM mode 3)
- $GTBER.CCRA[1:0] = 10b, 11b$  (GTCCRD performs buffer operation).

### TCFE flag (Input Compare Match Flag E)

The TCFE flag indicates the status for the compare match of GTCCRE.

When GTCCRE performs buffer operation, GTCCRE doesn't perform compare match.

[Setting condition]

- $GTCNT = GTCCRE$ .

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (GTCCRE performs buffer operation).

### TCFF flag (Input Compare Match Flag F)

The TCFF flag indicates the status for the compare match of GTCCRF.

When GTCCRF performs buffer operation, GTCCRF doesn't perform compare match.

[Setting condition]

- GTCNT = GTCCRF.

[Clearing condition]

- 0 is written to this flag.

[Not comparing condition]

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (Triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (GTCCRF performs buffer operation).

### TCFPO flag (Overflow Flag)

The TCFPO flag indicates when an overflow or crest has occurred.

[Setting conditions]

- In saw-wave mode, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred
- In triangle-wave mode, a crest (GTCNT changes from GTPR to GTPR - 1) has occurred
- In counting by hardware sources, an overflow (GTCNT changes from GTPR to 0 in up-counting) has occurred.

[Clearing condition]

- 0 is written to this flag.

### TCFPU flag (Underflow Flag)

The TCFPU flag indicates when an underflow or trough has occurred.

[Setting conditions]

- In saw-wave mode, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred
- In triangle-wave mode, a trough (GTCNT changes from 0 to 1) has occurred
- In counting by hardware sources, an underflow (GTCNT changes from 0 to GTPR in down-counting) has occurred.

[Clearing condition]

- 0 is written to this bit.

### TUCF flag (Count Direction Flag)

The TUCF flag indicates the count direction of GTCNT. In event count operation, this flag is set to 1 in up-counting and to 0 in down-counting.

### ADTRAUF flag (GTADTRA Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)

This status flag indicates generation of a GTADTRA register compare match in up-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRA register in up-counting.

[Clearing condition]

- 0 is written to the ADTRAUF flag.

**ADTRADF flag (GTADTRA Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)**

This status flag indicates generation of a GTADTRA register compare match in down-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRA register in down-counting.

[Clearing condition]

- 0 is written to the ADTRADF flag.

**ADTRBUF flag (GTADTRB Register Compare Match (Up-Counting) A/D Conversion Start Request Flag)**

This status flag indicates generation of a GTADTRB register compare match in up-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRB register in up-counting.

[Clearing condition]

- 0 is written to the ADTRBUF flag.

**ADTRBDF flag (GTADTRB Register Compare Match (Down-Counting) A/D Conversion Start Request Flag)**

This status flag indicates generation of a GTADTRB register compare match in down-counting.

[Setting condition]

- The GTCNT counter matches the GTADTRB register in down-counting.

[Clearing condition]

- 0 is written to the ADTRBDF flag.

**ODF flag (Output Disable Flag)**

The ODF flag shows the request of the output disable source group that is selected in the GRP[1:0] bits.

When output is disabled, an output disable control is not released within the same cycle in which an output disable request is negated. It is released in the next cycle.

**OABHF flag (Same Time Output Level High Flag)**

The OABHF flag indicates that the GTIOCnA pin and GTIOCnB pin output 1 at the same time.

When the GTIOCnA or GTIOCnB pin outputs 0, this flag returns to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When the output disable request by the OABHF flag is enabled (GTINTAD.GRPABH = 1), the OABHF flag is output to POEG as an output disable request. The GPT does not have an interrupt to indicate that outputs have been simultaneous driven to the high level. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 1 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCnA pin output value is different from the GTIOCnB pin output value when both OAE and OBE bits are set to 1
- The GTIOCnA and GTIOCnB pins output 0 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or OBE bit is set to 0.

**OABLF flag (Same Time Output Level Low Flag)**

The OABLF flag indicates that the GTIOCnA and GTIOCnB pins output 0 at the same time.

When the GTIOCnA pin or GTIOCnB pin outputs 1, this flag returns to 0. This flag is read only. Writing 0 to clear the flag is prohibited.

When the output disable request by the OABLF flag is enabled (GTINTAD.GRPABL = 1), the OABLF flag is output to POEG as an output disable request. The GPT does not have an interrupt to indicate that outputs have been simultaneous driven to the low level. Use the interrupt function in the POEG if this is necessary.

[Setting condition]

- The GTIOCnA and GTIOCnB pins output 0 at the same time when both OAE and OBE bits are set to 1.

[Clearing conditions]

- The GTIOCnA pin output value is different from the GTIOCnB pin output value when both OAE and OBE bits are set to 1
- The GTIOCnA and GTIOCnB pins output 1 at the same time when both OAE and OBE bits are set to 1
- Either the OAE bit or the OBE bit is set to 0.

The compare-target signals to generate the OABHF/OABLF flag are the compare match outputs (PWM outputs) signals before they are masked by the output disable function. Even during the output disable condition, compare match operation continues internally, where the OABHF or OABLF flag is updated based on the operation results.

### PCF flag (Period Count Function Finish Flag)

This bit is status flag of period count function finish.

[Setting condition]

- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 1 at the end of cycle.
- The GTPC.PCEN bit is 1 and the GTPC.PCNT counter is 0 at the count clock.

[Clearing condition]

- 0 is written to this bit.

## 21.2.17 GTBER : General PWM Timer Buffer Enable Register

Base address:  $GPT32n = 0x4032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT32n\_NS = 0x5032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT16m = 0x4032\_2000 + 0x0100 \times m$  (m = 10 to 13)  
 $GPT16m\_NS = 0x5032\_2000 + 0x0100 \times m$  (m = 10 to 13)

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ADTD B	ADTTB[1:0]	—	ADTD A	ADTTA[1:0]	—	CCRS WT	PR[1:0]	CCRB[1:0]	CCRA[1:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	BD2	BD1	BD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BD0	GTCCR Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
1	BD1	GTPR Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W



Bit	Symbol	Function	R/W
2	BD2	GTADTRA/GTADTRB Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
15:3	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CCRA[1:0]	GTCCRA Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTCCRA ↔ GTCCRC) Others: Double buffer operation (GTCCRA ↔ GTCCRC ↔ GTCCRD)	R/W
19:18	CCRB[1:0]	GTCCRB Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTCCRB ↔ GTCCRE) Others: Double buffer operation (GTCCRB ↔ GTCCRE ↔ GTCCRF)	R/W
21:20	PR[1:0]	GTPR Buffer Operation 0 0: No buffer operation 0 1: Single buffer operation (GTPBR → GTPR) Others: Setting prohibited	R/W
22	CCRSWT	GTCCRA and GTCCRB Forcible Buffer Operation Writing 1 to this bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after 1 is written. This bit is read as 0.	W
23	—	This bit is read as 0. The write value should be 0.	R/W
25:24	ADTTA[1:0]	GTADTRA Register Buffer Transfer Timing Select 0 0: In triangle wave mode, no transfer. In saw-wave mode, no transfer. 0 1: In triangle wave mode, transfer at crest. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 0: In triangle wave mode, transfer at trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 1: In triangle wave, transfer at both crest and trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing.	R/W
26	ADTDA	GTADTRA Register Double Buffer Operation 0: Single buffer operation (GTADTBRA → GTADTRA) 1: Double buffer operation (GTADTDBRA → GTADTBRA → GTADTRA)	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
29:28	ADTTB[1:0]	GTADTRB Register Buffer Transfer Timing Select 0 0: In triangle wave mode, no transfer. In saw-wave mode, no transfer. 0 1: In triangle wave mode, transfer at crest. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 0: In triangle wave mode, transfer at trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing. 1 1: In triangle wave mode, transfer at both crest and trough. In saw-wave mode, transfer at underflow (in down-counting), overflow (in up-counting), or counter clearing.	R/W
30	ADTDB	GTADTRB Register Double Buffer Operation 0: Single buffer operation (GTADTBRB → GTADTRB) 1: Double buffer operation (GTADTDBRB → GTADTBRB → GTADTRB)	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The GTBER register provides settings for the buffer operation. Set the GTBER register while the GTCNT counter is stopped.

#### BD0 bit (GTCCR Buffer Operation Disable)

The BD0 bit disables the buffer operation using GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, and GTCCRF combined.

When GTDTCR.TDE is 1 and when BD0 is set to 0, GTCCRB does not perform buffer operation. The GTCCRB register is automatically set to a compare match value for negative-phase waveform with dead time.

A value for the BD0 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the GTSECR.SBDCE or GTSECR.SBDCE.

#### **BD1 bit (GTPR Buffer Operation Disable)**

The BD1 bit disables the buffer operation using GTPR and GTPBR combined.

A value for the BD1 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the GTSECR.SBDPE or GTSECR.SBDPE.

#### **BD2 bit (GTADTRA/GTADTRB Registers Buffer Operation Disable)**

This bit disables buffer operation using the GTADTRA, GTADTBRA, and GTADTDBRA registers together and buffer operation using the GTADTRB, GTADTBRB, and GTADTDBRB registers together.

The setting is invalid during the event count operation, and the buffer operation using the GTADTRA and GTADTRB registers is not performed.

A value for the BD2 bit in the channel related to the position of the bit written with 1 by the GTSECSR register can be set when 1 is written to the SBDAD or SBDAD bit in the GTSECR register.

#### **CCRA[1:0] bits (GTCCRA Buffer Operation)**

The CCRA[1:0] bits set the buffer operation with GTCCRA, GTCCRC, and GTCCRD combined. When the buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.

The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough).

#### **CCRB[1:0] bits (GTCCRB Buffer Operation)**

The CCRB[1:0] bits set the buffer operation using GTCCRB, GTCCRE, and GTCCRF combined. When the buffer operation is restricted by the operating mode set in GTCR, the GTCR setting is given priority.

The buffer operation mode is fixed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough).

#### **PR[1:0] bits (GTPR Buffer Operation)**

The PR[1:0] bits set the buffer operation with GTPR and GTPBR combined.

#### **CCRSWT bit (GTCCRA and GTCCRB Forcible Buffer Operation)**

Writing 1 to the CCRSWT bit forces a buffer transfer of GTCCRA and GTCCRB. This bit automatically returns to 0 after the 1 is written. This bit is read as 0, and is valid only when counting is stopped with a compare match operation specified.

#### **ADTTA[1:0] bits (GTADTRA Register Buffer Transfer Timing Select)**

These bits set the transfer timing for buffer operation of the GTADTRA, GTADTBRA, and GTADTDBRA registers.

The setting is invalid during the event count operation.

#### **ADTDA bit (GTADTRA Register Double Buffer Operation)**

These bits set buffer operation with the GTADTRA, GTADTBRA, and GTADTDBRA registers combined.

The setting is invalid during the event count operation.

#### **ADTTB[1:0] bits (GTADTRB Register Buffer Transfer Timing Select)**

These bits set the transfer timing for buffer operation of the GTADTRB, GTADTBRB, and GTADTDBRB registers.

The setting is invalid during the event count operation.

#### **ADTDB bit (GTADTRB Register Double Buffer Operation)**

These bits set buffer operation with the GTADTRB, GTADTBRB, and GTADTDBRB registers combined.

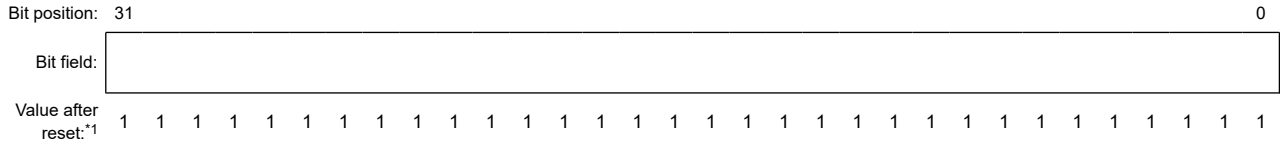
The setting is invalid during the event count operation.



### 21.2.20 GTPR : General PWM Timer Cycle Setting Register

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0x64



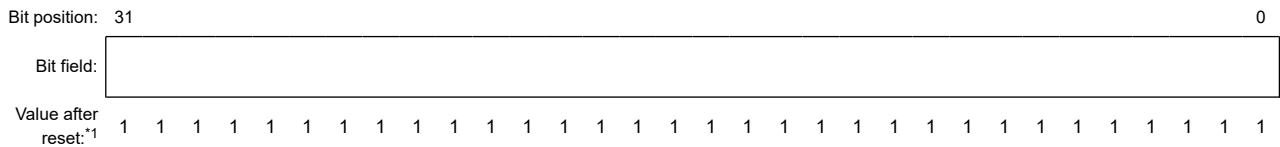
Bit	Symbol	Function	R/W
31:0	n/a	GTPR is a read/write register that sets the maximum count value of GTCNT. The effective size of GTPR is the same as GTCNT (16- or 32-bit). If the effective size of GTPR is 16 bits, the upper 16 bits for access in a 32-bit unit are always read as 0x0000, and writing to these bits is ignored. For saw waves, the value of (GTPR + 1) is the cycle. For triangle waves, the value of (GTPR value × 2) is the cycle.	R/W

Note 1. For GPT16m (m = 10 to 13), the value of the upper 16 bits after reset is 0x0000.

### 21.2.21 GTPBR : General PWM Timer Cycle Setting Buffer Register

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0x68



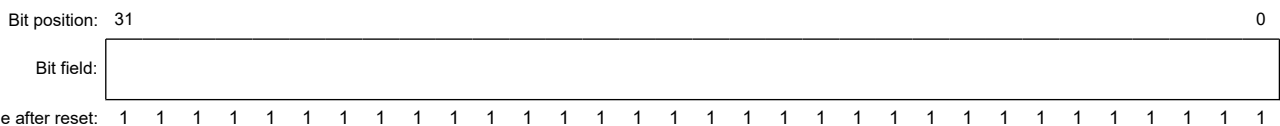
Bit	Symbol	Function	R/W
31:0	n/a	GTPBR is a read/write register that functions as a buffer register for GTPR. The effective size of GTPBR is the same as GTCNT (16- or 32-bit). If the effective size of GTPBR is 16 bits, the upper 16 bits for access in a 32-bit unit are always read as 0x0000, and writing to these bits is ignored.	R/W

Note 1. For GPT16m (m = 10 to 13), the value of the upper 16 bits after reset is 0x0000.

### 21.2.22 GTADTRk : A/D Conversion Start Request Timing Register k (k = A, B)

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)

Offset address: 0x70 (GTADTRA)  
 0x7C (GTADTRB)



Bit	Symbol	Function	R/W
31:0	n/a	Set the timing of A/D conversion start request generation Access in 8-bit or 16-bit units to the GTADTRk register is prohibited, and it should be accessed in 32-bit units. When the GTADTRk register value matches the GTCNT counter value, an A/D conversion start request is generated.	R/W

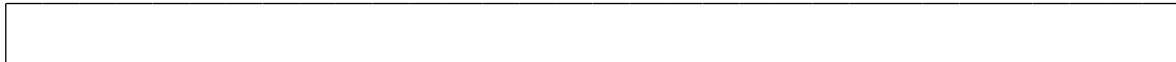
### 21.2.23 GTADTBRk : A/D Conversion Start Request Timing Buffer Register k (k = A, B)

Base address:  $GPT32n = 0x4032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT32n\_NS = 0x5032\_2000 + 0x0100 \times n$  (n = 0 to 5)

Offset address: 0x74 (GTADTBRA)  
 0x80 (GTADTBRB)

Bit position: 31 0

Bit field:



Value after reset: 1

Bit	Symbol	Function	R/W
31:0	n/a	The buffer registers for the GTADTRk register Access in 8-bit or 16-bit units to the GTADTBRk register is prohibited, and it should be accessed in 32-bit units.	R/W

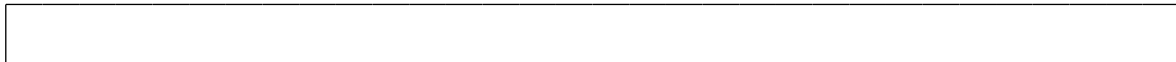
### 21.2.24 GTADTDBRk : A/D Conversion Start Request Timing Double-Buffer Register k (k = A, B)

Base address:  $GPT32n = 0x4032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT32n\_NS = 0x5032\_2000 + 0x0100 \times n$  (n = 0 to 5)

Offset address: 0x78 (GTADTDBRA)  
 0x84 (GTADTDBRB)

Bit position: 31 0

Bit field:



Value after reset: 1

Bit	Symbol	Function	R/W
31:0	n/a	The buffer registers for the GTADTBRk register (double buffer registers for the GTADTRk register) Access in 8-bit or 16-bit units to the GTADTDBRk register is prohibited, and it should be accessed in 32-bit units.	R/W

### 21.2.25 GTDTCR : General PWM Timer Dead Time Control Register

Base address:  $GPT32n = 0x4032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT32n\_NS = 0x5032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT16m = 0x4032\_2000 + 0x0100 \times m$  (m = 10 to 13)  
 $GPT16m\_NS = 0x5032\_2000 + 0x0100 \times m$  (m = 10 to 13)

Offset address: 0x88

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

Bit	Symbol	Function	R/W
0	TDE	Negative-Phase Waveform Setting 0: GTCCRB is set without using GTDVU 1: GTDVU is used to set the compare match value for negative-phase waveform with dead time automatically in GTCCRB	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

GTDTCR enables automatic setting of a compare match value for negative-phase waveform with dead time.

The setting is invalid during the event count operation.

#### TDE bit (Negative-Phase Waveform Setting)

The TDE bit specifies whether to use GTDVU. When GTDVU is used, the compare match value for a negative-phase waveform with dead time obtained by the compare match value of a positive-phase waveform (GTCCRA) and the dead time value (GTDVU) is automatically set in GTCCRB.

The TDE bit setting is ignored in saw-wave PWM mode, and the GTCCRB is not automatic setting.

The GTCCRB value is automatically set and has the following upper and lower limit values. If the obtained GTCCRB value is not within the upper or lower limit, the following limit value is set in GTCCRB.

- Triangle waves:  
 Upper limit value:  $GTPR - 1$   
 Lower limit value: 1 in up-counting, 0 in down-counting
- Saw-wave one-shot pulse mode:  
 Upper limit value:  $GTPR$   
 Lower limit value: 0.

### 21.2.26 GTDVU : General PWM Timer Dead Time Value Register U

Base address:  $GPT32n = 0x4032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT32n\_NS = 0x5032\_2000 + 0x0100 \times n$  (n = 0 to 5)  
 $GPT16m = 0x4032\_2000 + 0x0100 \times m$  (m = 10 to 13)  
 $GPT16m\_NS = 0x5032\_2000 + 0x0100 \times m$  (m = 10 to 13)

Offset address: 0x8C

Bit position:	31	0
Bit field:	[Empty box representing 32 bits]	
Value after reset:	1 1	

Bit	Symbol	Function	R/W
31:0	n/a	GTDVU is a read/write register that sets the dead time for generating PWM waveforms with dead time. The effective size of GTDVU is the same as GTCNT (16 or 32 bits). If the effective size of GTDVU is 16 bits, the upper 16 bits for access in a 32-bit unit are always read as 0x0000, and writing to these bits is ignored. Setting a GTDVU value greater than or equal to GTPR is prohibited. When using the automatic dead time setting function, do not set a value that makes a change point of the waveform exceeding the count period. The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. When GTDVU is used, writing to GTCCRB is prohibited. When this register is set to 0, waveforms without dead time are output.	R/W

Note 1. For GPT16m (m = 10 to 13), the value of the upper 16 bits after reset is 0x0000.

### 21.2.27 GTADSMR : General PWM Timer A/D Conversion Start Request Signal Monitoring Register

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0xA4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ADSM EN1	—	—	—	—	—	—	—	ADSMS1[1:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADSM EN0	—	—	—	—	—	—	—	ADSMS0[1:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ADSMS0[1:0]	A/D Conversion Start Request Signal Monitor 0 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting. 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting. 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting. 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	ADSMEN0	A/D Conversion Start Request Signal Monitor 0 Output Enabling 0: Output of A/D conversion start request signal monitor 0 is disabled. 1: Output of A/D conversion start request signal monitor 0 is enabled.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
17:16	ADSMS1[1:0]	A/D Conversion Start Request Signal Monitor 1 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting. 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting. 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting. 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting.	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	ADSMEN1	A/D Conversion Start Request Signal Monitor 1 Output Enabling 0: Output of A/D conversion start request signal monitor 1 is disabled. 1: Output of A/D conversion start request signal monitor 1 is enabled.	R/W

Bit	Symbol	Function	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The GTADSMR register is used to control monitors for the A/D conversion start request signal that is synchronized with a frame period.

**ADSMSk[1:0] bits (A/D Conversion Start Request Signal Monitor k Selection) (k = 0, 1)**

These bits are used to select A/D conversion start request signal synchronized with a frame period which is monitored by the GTASMk pin.

In triangle-wave PWM mode, the following settings are prohibited:

- Set ADSMSk[1:0] bit to 00b (A/D conversion start request during up-counting) when GTADTRA = 0
- Set ADSMSk[1:0] bit to 10b (A/D conversion start request during up-counting) when GTADTRB = 0
- Set ADSMSk[1:0] bit to 01b (A/D conversion start request during down-counting) when GTADTRA = GTPR
- Set ADSMSk[1:0] bit to 11b (A/D conversion start request during down-counting) when GTADTRB = GTPR

**ADSMENk bit (A/D Conversion Start Request Signal Monitor k Output Enabling) (k = 0, 1)**

This bit enables or disables the monitor output to the GTADSMk pin.

When the output is disabled, the GTADSMk pin goes to the low level.

When the bit is 1, the signal on the GTADSMk pin goes to the high level on assertion of the signal to request to the start of A/D conversion selected by the ADSMSk[1:0] bits and returns to the low level at the end of the current cycle of the timer for the channel that generated the given signal to request the start of A/D conversion. When the counter stops, the value when the counter stopped is retained for output. Set the ADSMENk bit to 0 to output the low level.

When a signal to request the start of A/D conversion is generated at the end of a timer period, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next period.

When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPT.

**21.2.28 GTICLF : General PWM Timer Inter Channel Logical Operation Function Setting Register**

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0xB8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ICLFSELD[5:0]					—	ICLFB[2:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ICLFSELC[5:0]					—	ICLFA[2:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Bit	Symbol	Function	R/W
2:0	ICLFA[2:0]	GTIOCnA Output Logical Operation Function Select 0 0 0: A (no delay) 0 0 1: NOT A (no delay) 0 1 0: C (1 PCLKD delay) 0 1 1: NOT C (1 PCLKD delay) 1 0 0: A AND C (1 PCLKD delay) <sup>*2</sup> 1 0 1: A OR C (1 PCLKD delay) <sup>*2</sup> 1 1 0: A EXOR C (1 PCLKD delay) <sup>*2</sup> 1 1 1: A NOR C (1 PCLKD delay) <sup>*2</sup>	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
9:4	ICLFSEL[5:0]	Inter Channel Signal C Select <sup>*1*2</sup> 0x00: GTIOC0A 0x01: GTIOC0B 0x02: GTIOC1A 0x03: GTIOC1B : : 0x0A: GTIOC5A 0x0B: GTIOC5B 0x14: GTIOC10A 0x15: GTIOC10B : : 0x1A: GTIOC13A 0x1B: GTIOC13B Others: Setting prohibited	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
18:16	ICLFB[2:0]	GTIOCnB Output Logical Operation Function Select 0 0 0: B (no delay) 0 0 1: NOT B (no delay) 0 1 0: D (1 PCLKD delay) 0 1 1: NOT D (1 PCLKD delay) 1 0 0: B AND D (1 PCLKD delay) <sup>*3</sup> 1 0 1: B OR D (1 PCLKD delay) <sup>*3</sup> 1 1 0: B EXOR D (1 PCLKD delay) <sup>*3</sup> 1 1 1: B NOR D (1 PCLKD delay) <sup>*3</sup>	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
25:20	ICLSEL[5:0]	Inter Channel Signal D Select <sup>*1*3</sup> 0x00: GTIOC0A 0x01: GTIOC0B 0x02: GTIOC1A 0x03: GTIOC1B : : 0x0A: GTIOC5A 0x0B: GTIOC5B 0x14: GTIOC10A 0x15: GTIOC10B : : 0x1A: GTIOC13A 0x1B: GTIOC13B Others: Setting prohibited	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: n = 0 to 5, 10 to 13

Note 1. The signal before performing output disable control is selected.

Note 2. When channel's own GTIOCnA is selected, C is treated as "1".

Note 3. When channel's own GTIOCnB is selected, D is treated as "1".

The GTICLF register sets the logical operation function between compare match outputs. The logical operation is performed with the signals that the duty 0%/100% control is performed after compare match control. (The output disable control is performed with the signal after logical operation.)

Access in 8-bit units to GTICLF is prohibited.

**ICLFm[2:0] bit (GTIOCnm Output Logical Operation Function Select) (m = A, B)**

These bits select the logical operation function between signals before performing output disable control for GTIOCnm. To prevent hazard to the GPT output, the signal after logical operation is latched with PCLKD. After latching, the output disable control is performed. When the logical operation function which causes the delay of 1 PCLKD is selected, the output enable signal is also delayed with 1 PCLKD and input to the output disable control.

When the same signal to operate logical function AND, OR, EXOR and NOR is selected, one signal is treated as “1”.

**ICLFSELk[5:0] bit (Inter Channel Signal k Select) (k = C, D)**

These bits select the signal k that the logical operation is performed with the signal before performing output disable control for GTIOCnm.

**21.2.29 GTPC : General PWM Timer Period Count Register**

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 3)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 3)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10)

Offset address: 0xBC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	PCNT[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ASTP	—	—	—	—	—	—	—	PCEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PCEN	Period Count Function Enable 0: Period count function is disabled 1: Period count function is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	ASTP	Automatic Stop Function Enable 0: Automatic stop function is disabled 1: Automatic stop function is enabled	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
27:16	PCNT[11:0]	Period Counter Counter for the number of period	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The GTPC register counts the number of period.

**PCEN bit (Period Count Function Enable)**

This bit enables or disables period count function.

Writing is available when counting is both in progress and stopped.

When 1 is written to either the GTSECR.SPCE bit or the GTSECR.SPCD bit, the value is simultaneously set to the PCEN bit in the channels set to 1 by the GTSECSR register.

**ASTP bit (Automatic Stop Function Enable)**

This bit enables or disables the GTCNT counter automatic stopping after finishing counting the number of period.

When the PCEN bis is 0, writing is available.

When the PCEN bit is 1, writing is disabled.

When the PCEN bit is 1, the ASTP bit is 1, and the PCNT counter is stopped at PCNT = 0, the GTCNT counter is also stopped. When the ASTP bit is 0, the GTCNT counter continues to count.

**PCNT[11:0] bits (Period Counter)**

This counter counts the number of period.

When the PCEN bit is 0, writing the number of period is available.

When the PCEN bit is 1, writing is disabled, and down-counting is performed at the end of period. In saw-wave mode, the end of period refers to overflow, underflow, or counter clearing. In triangle-wave mode, it refers to trough.

When the PCNT counter is 1 at the end of period, it becomes 0 and counting is stopped.

When the GTCNT counter is stopped while period count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value.

When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

**21.2.30 GTSECSR : General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register**

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0xD0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	SECS EL13	SECS EL12	SECS EL11	SECS EL10	—	—	—	—	SECS EL5	SECS EL4	SECS EL3	SECS EL2	SECS EL1	SECS EL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SECSEL0	Channel 0 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
1	SECSEL1	Channel 1 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
2	SECSEL2	Channel 2 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
3	SECSEL3	Channel 3 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
4	SECSEL4	Channel 4 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
5	SECSEL5	Channel 5 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
9:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10	SECSEL10	Channel 10 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
11	SECSEL11	Channel 11 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
12	SECSEL12	Channel 12 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
13	SECSEL13	Channel 13 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The GTSECSR register selects an intended channel n (n = 0 to 5, 10 to 13) for updating an operation enable bit by the GTSECR register. A bit position for the GTSECSR register indicates a channel number. The GTSECSR register of each channel is a common register, and writing 1 to a bit in the GTSECSR register in any channel and updating it changes a channel, related to the position of the bit written with 1 by the GTSECSR register, to be simultaneously controlled of the operation enable bit by the GTSECR register.

Contrary to a security attribute for write or read access, if the security and privilege attributes set on each channel has the security violation or the privilege violation, the bit of the channel number that violates security cannot be written or read, and the read value of the bit is "0".

For the association between the GTSECSR bit number and a channel number, see [Figure 21.2](#).

Access in 8-bit or 16-bit units to GTSECSR is prohibited, and it should be accessed in 32-bit units.

**SECSELn bit (Operation Enable Bit Simultaneous Control Channel Select) (n = 0 to 5, 10 to 13)**

This bit enables or disables the simultaneous control of operation enable in channel n.

When the bit is set to 1, the simultaneous control is enabled, and disabled when the bit is 0.

**21.2.31 GTSECR : General PWM Timer Operation Enable Bit Simultaneous Control Register**

Base address: GPT32n = 0x4032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT32n\_NS = 0x5032\_2000 + 0x0100 × n (n = 0 to 5)  
 GPT16m = 0x4032\_2000 + 0x0100 × m (m = 10 to 13)  
 GPT16m\_NS = 0x5032\_2000 + 0x0100 × m (m = 10 to 13)

Offset address: 0xD4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	SPCD	—	—	—	—	—	—	—	SPCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SBDA D	SBDP D	SBDC D	—	—	—	—	—	SBDA E	SBDP E	SBDC E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SBDCE	GTCCR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTCCR buffer operations 1: Enable GTCCR register buffer operations simultaneously	R/W
1	SBDPE	GTPR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTPR buffer operations 1: Enable GTPR register buffer operations simultaneously	R/W

Bit	Symbol	Function	R/W
2	SBDAE	GTADTR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTADTR buffer operations 1: Enable GTADTR register buffer operations simultaneously	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	SBDCE	GTCCR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTCCR buffer operations 1: Disable GTCCR register buffer operations simultaneously	R/W
9	SBDPE	GTPR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTPR buffer operations 1: Disable GTPR register buffer operations simultaneously	R/W
10	SBDAD	GTADTR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTADTR buffer operations 1: Disable GTADTR register buffer operations simultaneously	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	SPCE	Period Count Function Simultaneous Enable*1 0: Disable simultaneous enabling period count function 1: Enable period count function simultaneously	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	SPCD	Period Count Function Simultaneous Disable*1 0: Disable simultaneous disabling period count function 1: Disable period count function simultaneously	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is only available in GPT320 to GPT323, GPT1610.

In GPT324, GPT325, GPT1611 to GPT1613, this bit is read as 0. The write value should be 0.

The GTSECR register simultaneously updates the value for operation enable bits of a channel set by the GTSECSR register. Writing 1 to a bit in the GTSECR register in any channel and updating it updates an operation enable bit for all channels, related to the position of the bit written with 1 by the all GTSECSR registers.

Setting enable and disable bits for the same operation enable bit to 1 in the GTSECR is prohibited.

Contrary to a security attribute for write or read access, if the security and privilege attributes set on each channel has the security violation or the privilege violation, the bit of the channel number that violates security cannot be written or read, and the read value of the bit is "0".

A bit written to 1 is automatically cleared. When the GTSECR is read, 0 is read.

Access in 8-bit or 16-bit units to the GTSECR register is prohibited, and it should be accessed in 32-bit units.

#### **SBDCE bit (GTCCR Register Buffer Operation Simultaneous Enable)**

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are enabled.

Simultaneous setting of SBDCE and SBDCE bits to 1 is prohibited.

#### **SBDPE bit (GTPR Register Buffer Operation Simultaneous Enable)**

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR and GTPBR registers are enabled.

Simultaneous setting of SBDPE and SBDPE bits to 1 is prohibited.

#### **SBDAE bit (GTADTR Register Buffer Operation Simultaneous Enable)**

When 1 is written to this bit, 0 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTRBR, and GTADTDBRB registers are enabled.

Simultaneous setting of SBDAE and SBDAD bits to 1 is prohibited.

**SBDCE bit (GTCCR Register Buffer Operation Simultaneous Disable)**

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[0] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTCCRA, GTCCRC, and GTCCRD registers and using the GTCCRB, GTCCRE, and GTCCRF registers are disabled.

Simultaneous setting of SBDCE and SBDCE bits to 1 is prohibited.

**SBDPE bit (GTPR Register Buffer Operation Simultaneous Disable)**

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[1] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTPR and GTPBR registers are disabled.

Simultaneous setting of SBDPE and SBDPE bits to 1 is prohibited.

**SBDAD bit (GTADTR Register Buffer Operation Simultaneous Disable)**

When 1 is written to this bit, 1 is simultaneously set to a GTBER.BD[2] bit in the channels set to 1 by the GTSECSR register, and buffer operations using the GTADTRA, GTADTBRA, and GTADTDBRA registers and using the GTADTRB, GTADTRBR, and GTADTDBRB registers are disabled.

Simultaneous setting of SBDAD and SBDAD bits to 1 is prohibited.

**SPCE bit (Period Count Function Simultaneous Enable)**

When 1 is written to this bit, 1 is simultaneously set to GTPC.PCEN bit in the channels set to 1 by the GTSECSR register, and period count function is enabled.

Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

**SPCD bit (Period Count Function Simultaneous Disable)**

When 1 is written to this bit, 0 is simultaneously set to GTPC.PCEN bit in the channels set to 1 by the GTSECSR register, and period count function is disabled.

Simultaneous setting of SPCE and SPCD bits to 1 is prohibited.

## 21.3 Operation

### 21.3.1 Basic Operation

Each channel has a 32-bit and 16-bit timer that performs a periodic count operation using the count clock and hardware sources. The count function provides both up-counting and down-counting. The GTPR controls the count cycle.

When the GTCNT counter value matches the value in GTCCRA or GTCCRB, the output from the associated GTIOCnA or GTIOCnB can be changed (n = 0 to 5, 10 to 13). GTCCRA or GTCCRB can be used as an input capture register with hardware resources.

GTCCRC and GTCCRD can function as buffer registers for GTCCRA. GTCCRE and GTCCRF can function as buffer registers for GTCCRB.

#### 21.3.1.1 Counter operation

##### (1) Counter start and stop

The counter of each channel starts the count operation when GTCR.CST is set to 1, and stops counting when the bit is set to 0. The GTCR.CST bit value is changed by the following sources:

- Writing to GTCR register
- Writing 1 to the bit in GTSTR associated with the GPT channel number when the GTSSR.CSTRT bit set to 1
- Writing 1 to the bit in GTSTP associated with the GPT channel number when the GTPSR.CSTOP bit set to 1
- The hardware source selected in the GTSSR register
- The hardware source selected in the GTPSR register
- Completion of the period count function while the GTPC.ASTP bit is 1

(2) Periodic count operation in up-counting by count clock

The GTCNT counter in each channel starts up-counting when the associated GTCR.CST bit is set to 1 with GTUPSR and GTDNSR registers set to 0x00000000. When the GTCNT value changes from the GTPR value to 0 (overflow), the GTST.TCFPO flag is set to 1, and the overflow interrupt (GPTn\_OVF) is also generated. After GTCNT overflows, up-counting resumes from 0x00000000.

Figure 21.3 shows an example of a periodic count operation in up-counting by the count clock.

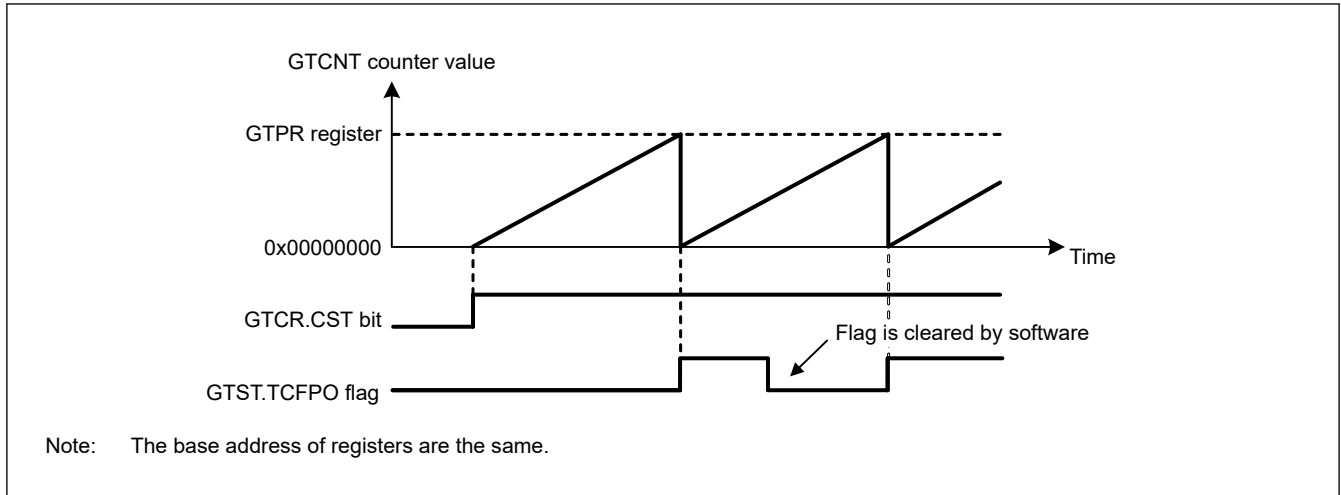


Figure 21.3 Example of periodic count operation in up-counting by the count clock

Table 21.5 shows an example for setting periodic count operation in up-counting by the count clock.

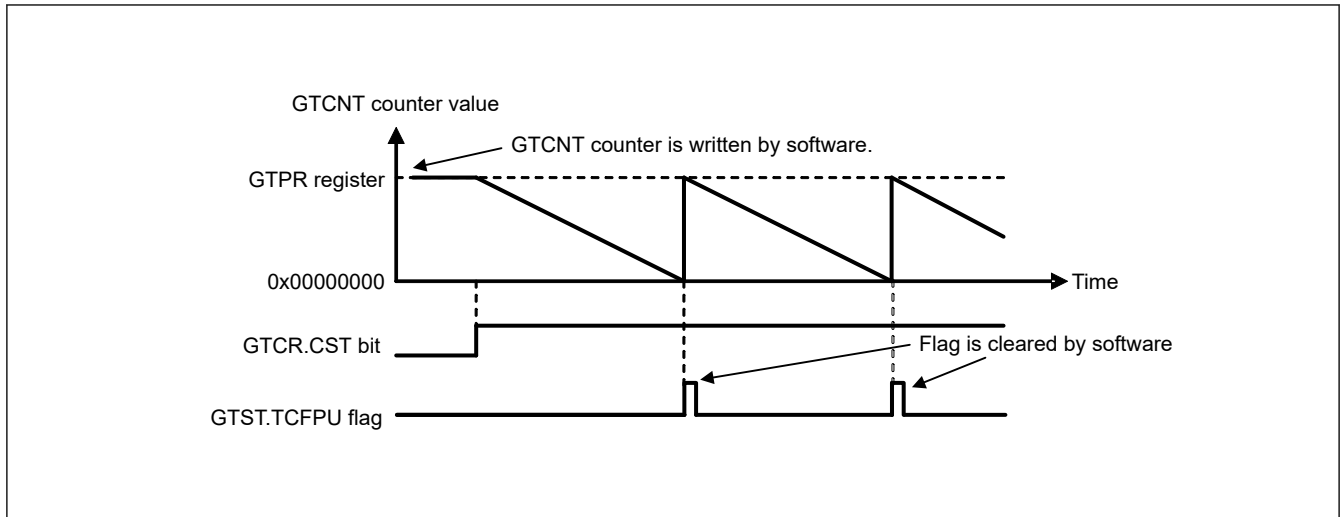
Table 21.5 Example for setting a periodic count operation in up-counting by the count clock

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.3, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.3, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.3, 0x00000000 is set.
6	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

(3) Periodic count operation in down-counting by count clock

The GTCNT counter in each channel can perform down-counting by setting GTUDDTYC.UD with GTUPSR and GTDNSR registers set to 0x00000000. When GTCNT changes from 0 to the GTPR value (underflow), GTST.TCFPU is set to 1, and the underflow interrupt(GPTn\_UDF) is also generated. After the GTCNT counter underflows, down-counting resumes from the GTPR value.

Figure 21.4 shows an example of periodic count operation in down-counting by the count clock.



**Figure 21.4 Example of periodic count operation in down-counting by the count clock**

Table 21.6 shows an example for setting periodic count operation in down-counting by the count clock.

**Table 21.6 Example for setting periodic count operation in down-counting by count clock**

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.4, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction with the GTUDDTYC register. In Figure 21.4, after 10b is set in the GTUDDTYC[1:0] bits, 00b is set in the GTUDDTYC[1:0] bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.4, the GTPR register value is set.
6	Start count operation	Set the GTCR.CST bit to 1 to start count operation. In Figure 21.4, 1 is set in the CST bit.

#### (4) Event count operation in up-counting using hardware sources

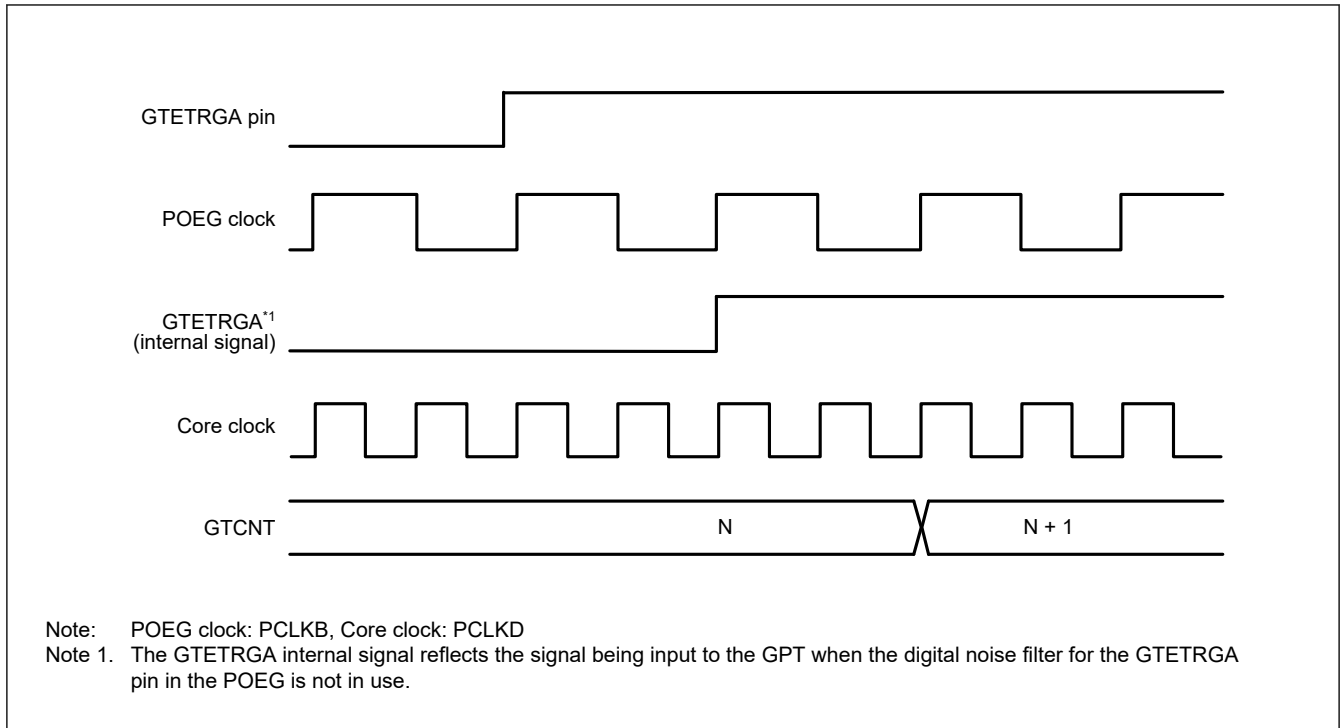
The GTCNT counter in each channel can perform up-counting using hardware sources as set in GTUPSR.

When GTUPSR is set to enable, the count clock selected in GTCR.TPCS[3:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, the GTCNT counter value does not change. The overflow behavior when up-counting using hardware sources is the same as when up-counting by the count clock.

If you are using a hardware source to count up, set the GTCR.CST bit to 1 to enable the counting operation. After GTCR.CST is set to 1, the counter cannot count up for 1 clock cycle as specified in GTCR.TPCS[3:0] because the count operation is synchronized by the count clock selected in GTCR.TPCS[3:0]. Set GTCR.TPCS[3:0] to 000b to count up with a 1 PCLKD delay after GTCR.CST is set to 1.

Figure 21.5 shows an example of an event count operation in up-counting by a hardware resource (the rising edge of GTETRGA pin input).





**Figure 21.5 Example of event count operation in up-counting using hardware sources**

Table 21.7 shows an example for setting event count operation in up-counting by a hardware source.

**Table 21.7 Example for setting an event count operation in up-counting using hardware sources**

No.	Step Name	Description
1	Set count source	Select the counting-up hardware source with the GTUPSR register.
2	Set cycle	Set the cycle in the GTPR register.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

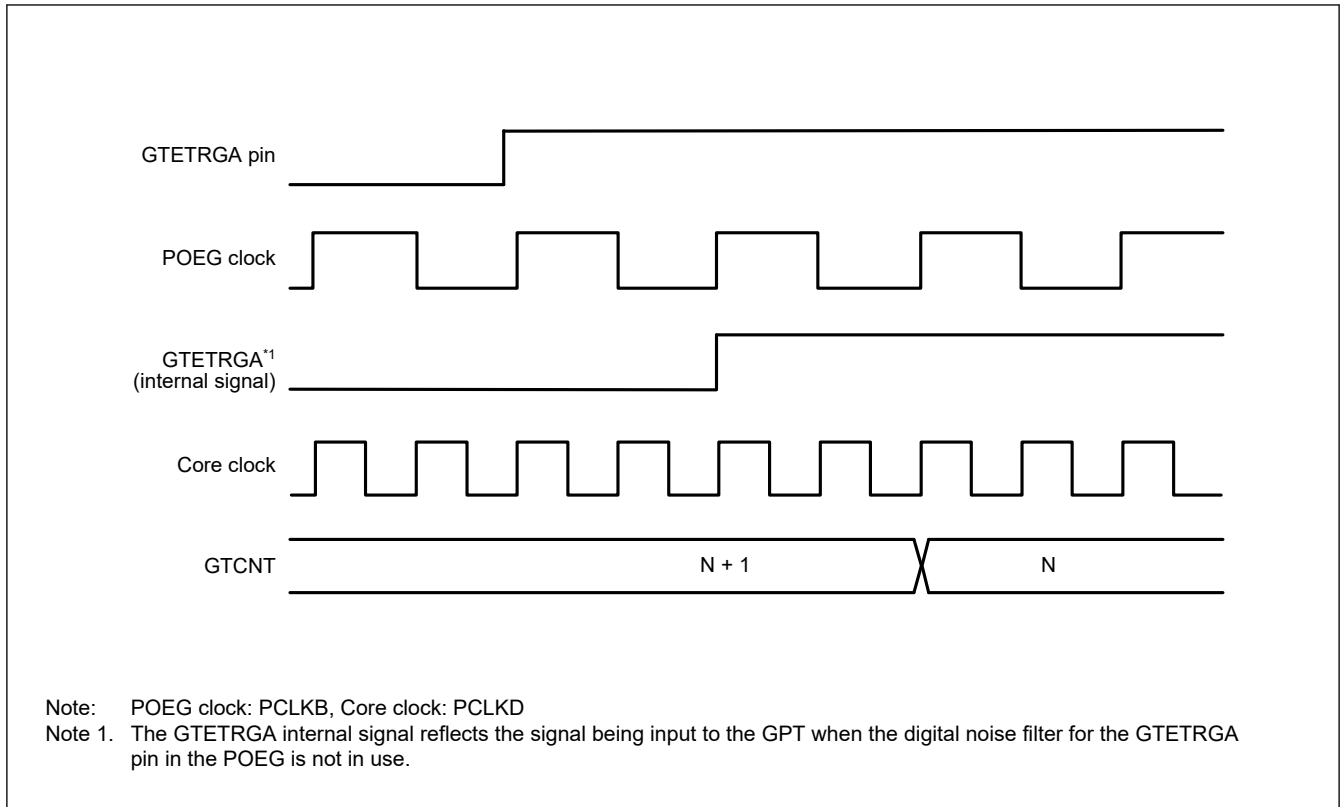
**(5) Event count operation in down-counting using hardware sources**

The GTCNT counter in each channel can perform down-counting using hardware sources set in the GTDNSR.

When GTDNSR is set to enable, the count clock selected in GTCR.TPCS[3:0] and the count direction selected in GTUDDTYC.UD are ignored. If up-counting and down-counting using hardware sources occur at the same time, the GTCNT counter value does not change. The underflow behavior when down-counting using hardware sources is the same as when down-counting by the count clock.

When GTCR.CST bit is set to 1 to count down using hardware sources, the count operation is enabled. After GTCR.CST is set to 1, the counter cannot count down for 1 clock cycle as specified in GTCR.TPCS[3:0] because the count operation is synchronized with the count clock selected in GTCR.TPCS[3:0]. Set GTCR.TPCS[3:0] to 000b to count down with a 1 PCLKD delay after GTCR.CST is set to 1.

Figure 21.6 shows an example of a event count operation in down-counting by a hardware resource (rising edge of GTETRGA pin).



**Figure 21.6 Example of event count operation in down-counting using hardware sources**

Table 21.8 shows an example for setting a periodic count operation in down-counting using a hardware resource.

**Table 21.8 Example for setting an event count operation in down-counting using hardware sources**

No.	Step Name	Description
1	Set count source	Select the counting-down hardware source with the GTDNSR register.
2	Set cycle	Set the cycle in the GTPR register.
3	Set initial value for counter	Set the initial value in the GTCNT counter.
4	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

### (6) Counter clear operation

The counter of each channel is cleared by following sources:

- Writing 0 to GTCNT register
- Writing 1 to the bit in GTCLR associated with the GPT channel number when the GTCSR.CCLR bit set to 1
- The hardware source selected in GTCSR register.

Writing to the GTCNT register is prohibited during count operation. The GTCNT counter can be cleared both by writing 1 to the GTCLR and by the clear request of hardware sources, whether GTCNT is counting (GTCR.CST is 1) or not (GTCR.CST is 0).

When the count direction flag is set as decrement (GTST.TCUF flag = 0) in saw-wave mode selected with GTCR.MD[2:0] bits, the GTCNT register is set to the value of the GTPR register when writing 1 to the GTCLR register and when clearing by hardware sources are performed.

When not in saw-waves mode and down-counting, the GTCNT register is set to 0 when writing 1 to the GTCLR register and when clearing by hardware sources are performed.

In event count operation when at least 1 bit in the GTUPSR or GTDNSR is set to 1, after clear sources occur, both writing to GTCLR register and clearing by hardware sources are performed immediately to synchronize with PCLKD. If other settings are used, clear is synchronized with the counter clock selected in GTCR.TPCS[3:0].

### 21.3.1.2 Waveform output by compare match

Compare match means that the GTCNT counter value matches the value of GTCCRA or GTCCRB. When a compare match occurs, the compare match flag is generated synchronously with the count clock, including the event count. At the same time, the GPT can output low, high, or toggled output from the associated GTIOCnA or GTIOCnB output pin (n = 0 to 5, 10 to 13). In addition, the GTIOCnA or GTIOCnB pin output can be low, high, or toggled at the cycle end which is determined by GTPR.

The cycle end is:

- For saw waves in up-counting – when GTCNT changes from the GTPR value to 0 (overflow)
- For saw waves in down-counting – when GTCNT changes from 0 to GTPR value (underflow)
- For saw waves – when the GTCNT counter is cleared
- For triangle waves – when the GTCNT changes from 0 to 1 (trough).

#### (1) Low output and high output

Figure 21.7 shows an example of low output and high output operation by a compare match of GTCCRA and GTCCRB.

In this example, the GTCNT counter performs up-counting, and settings are made so that high is output from the GTIOCnA pin by a GTCCRA compare match, and low is output from the GTIOCnB pin by a GTCCRB compare match. The pin level does not change when the specified level and pin level match.

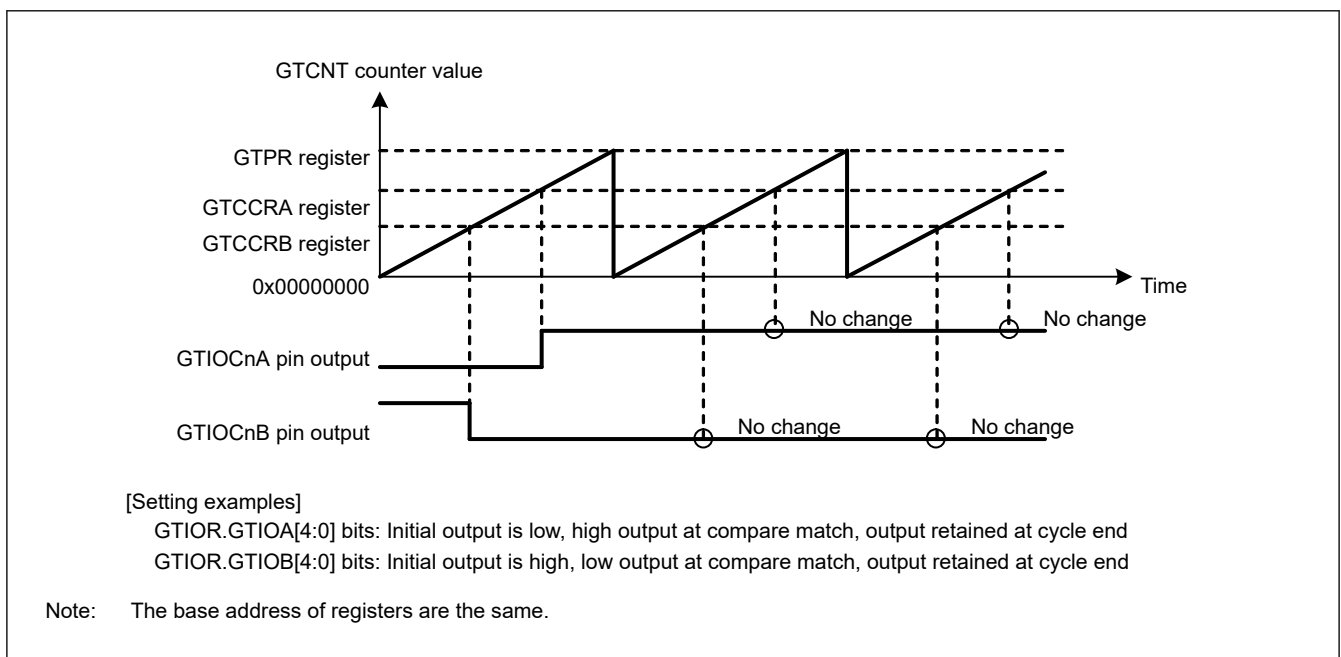


Figure 21.7 Example of low output and high output operation

Table 21.9 shows an example for setting low output and high output operation.

Table 21.9 Example for setting low output and high output operation (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.7, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.7, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.

**Table 21.9 Example for setting low output and high output operation (2 of 2)**

No.	Step Name	Description
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In <a href="#">Figure 21.7</a> , GTIOA[4:0] = 00010b, GTIOB[4:0] = 10001b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value	Set compare match values in the GTCCRA and GTCCRB registers.
9	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

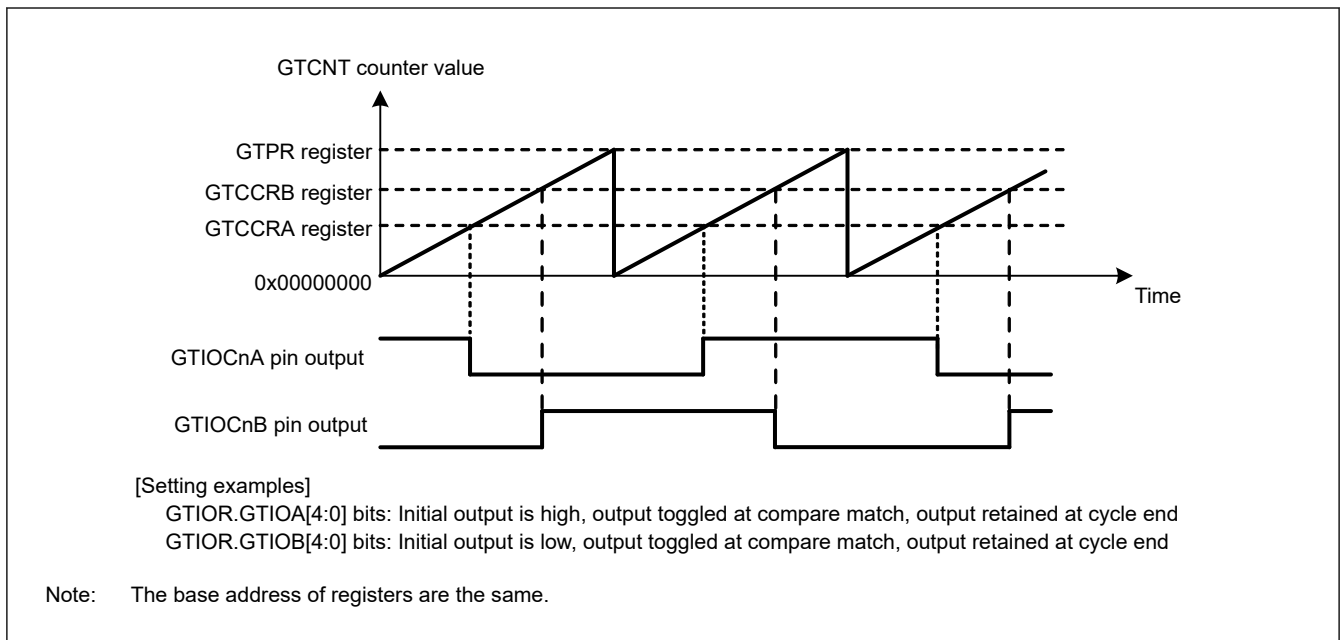
Note: n: 0 to 5, 10 to 13  
m: A, B

(2) Toggled output

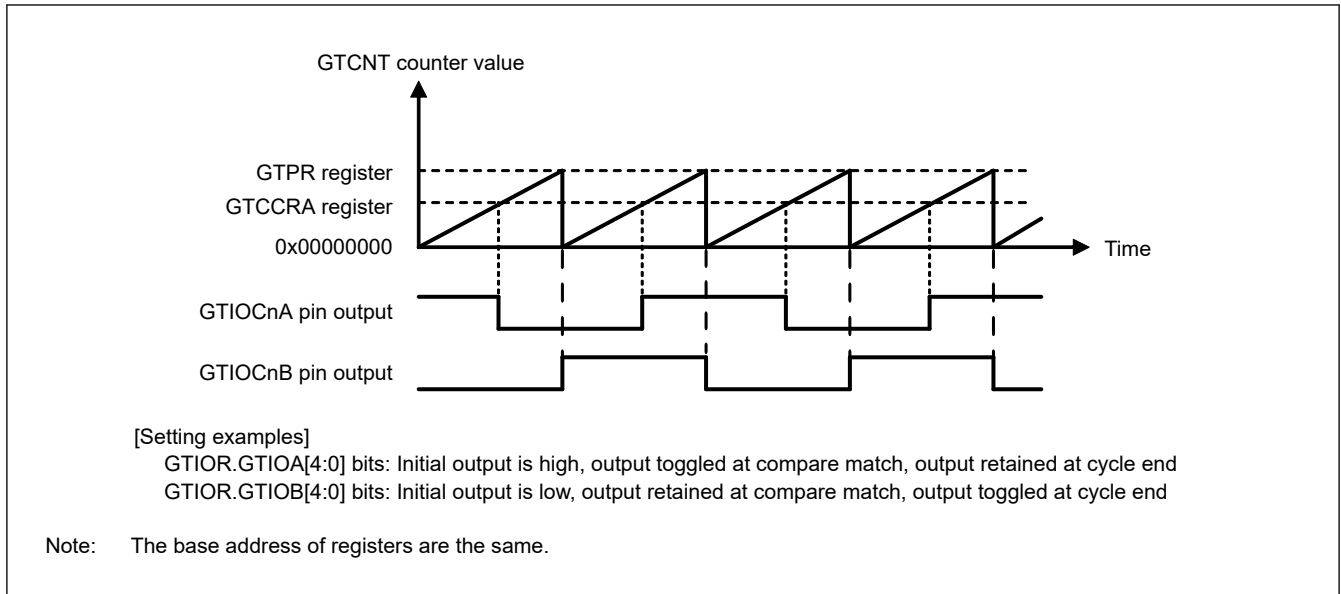
[Figure 21.8](#) and [Figure 21.9](#) show examples of toggled output operation by compare matches of GTCCRA and GTCCRB.

In [Figure 21.8](#), the GTCNT counter performs up-counting, and settings are made so that the GTIOCnA pin output by a GTCCRA compare match and GTIOCnB pin output by a GTCCRB compare match are toggled.

In [Figure 21.9](#), the GTCNT counter performs up-counting, and settings are made so that a GTCCRA compare match toggles the GTIOCnA pin output level and a cycle end toggles the GTIOCnB pin output level.



**Figure 21.8 Example of toggled output operation (1)**



**Figure 21.9 Example of toggled output operation (2)**

Table 21.10 shows an example for setting toggled output operation.

**Table 21.10 Example for setting toggled output operation**

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.8 and Figure 21.9, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.8 and Figure 21.9, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOcNm pin function	Set the GTIOcNm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.8, GTIOA[4:0] = 10011b, GTIOB[4:0] = 00011b, and in Figure 21.9, GTIOA[4:0] = 10011b, GTIOB[4:0] = 01100b.
7	Enable GTIOcNm pin output	Set to enable the GTIOcNm pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value	Set compare match values in the GTCCRA and GTCCRB registers.
9	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

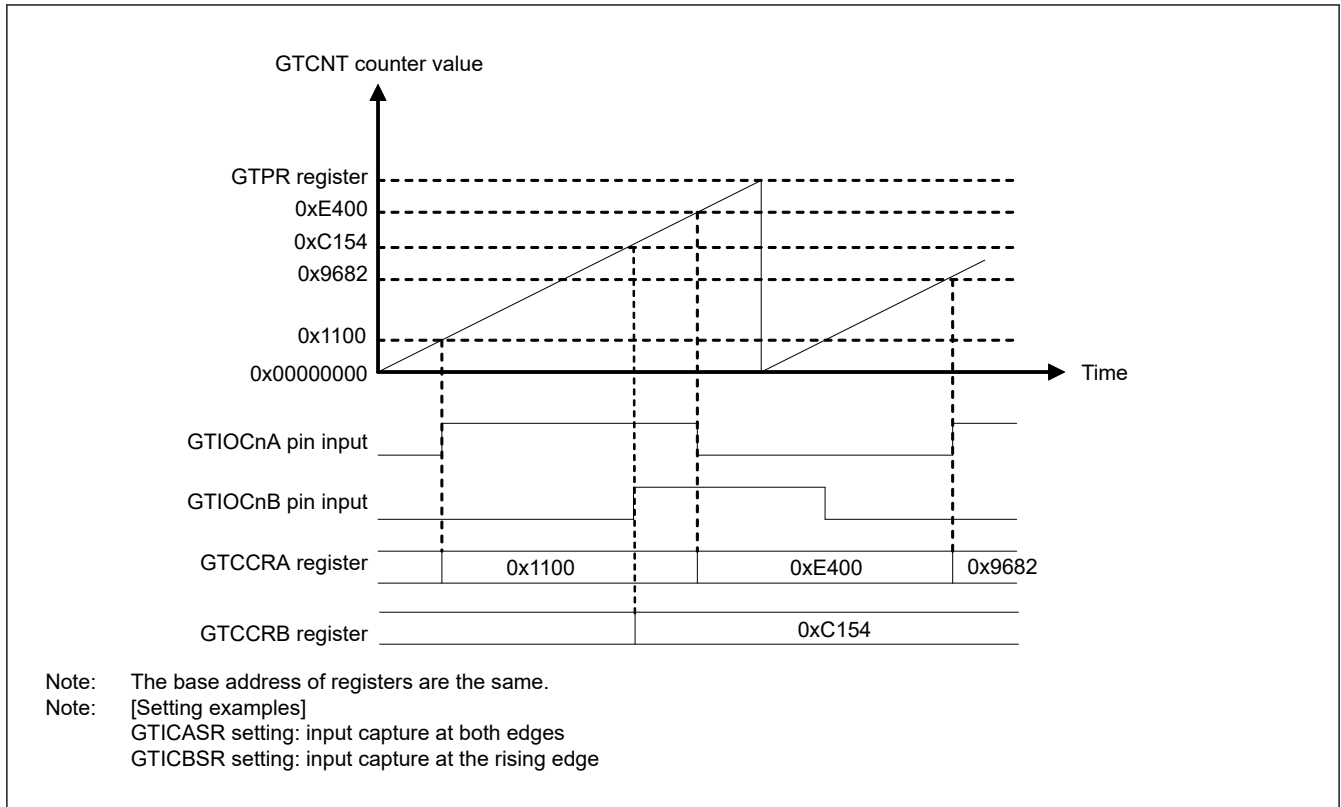
Note: n: 0 to 5, 10 to 13  
 m: A, B

### 21.3.1.3 Input Capture Function

The GTCNT counter value can be transferred to either GTCCRA or GTCCRB on detection of the hardware source that is set in GTICASR and GTICBSR.

Figure 21.10 shows an example of the input capture function.

In this example, the GTCNT counter performs up-counting by the count clock, and settings are made so that an input capture is performed to GTCCRA at both edges of the GTIOcNA input pin and to GTCCRB on the rising edge of the GTIOcNB input pin.



**Figure 21.10 Example of input capture operation**

Table 21.11 and Table 21.14 show the example for setting an input capture operation with count operation by the count clock.

**Table 21.11 Example for setting input capture operation**

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.10, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.10, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Select input capture source	Select the input capture source in the GTICASR and GTICBSR registers. In Figure 21.10, GTICASR = 0x00000F00, GTICBSR = 0x00003000.
7	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

### 21.3.2 Buffer Operation

The following buffer operations can be set with GTBER:

- GTPR and GTPBR
- GTCCRA, GTCCRC, and GTCCRD
- GTCCRB, GTCCRE, and GTCCRF
- GTADTRA, GTADTBRA, and GTADTDBRA
- GTADTRB, GTADTBRB, and GTADTDBRB

### 21.3.2.1 GTPR Register Buffer Operation

GTPBR can function as a buffer register for GTPR.

The buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count, and at a trough in triangle-wave mode.

In saw-wave mode or in event count, the buffer transfer is performed when the following counter clear operations occur during counting:

- Clear by hardware sources (the clear source is selected in GTCSR register)
- Clear by software (when GTCSR.CCLR bit is 1 and GTCLR.CCLRn bit is set to 1, n = 0 to 5, 10 to 13).

To set GTPR to function as a buffer, set the GTBER.PR bit to 1. To set GTPR not to function as a buffer, set the GTBER.PR bit to 0.

Figure 21.11 to section 21.3.2.1. GTPR Register Buffer Operation show examples of GTPR buffer operation and Table 21.12 shows an example for setting GTPR buffer operation.

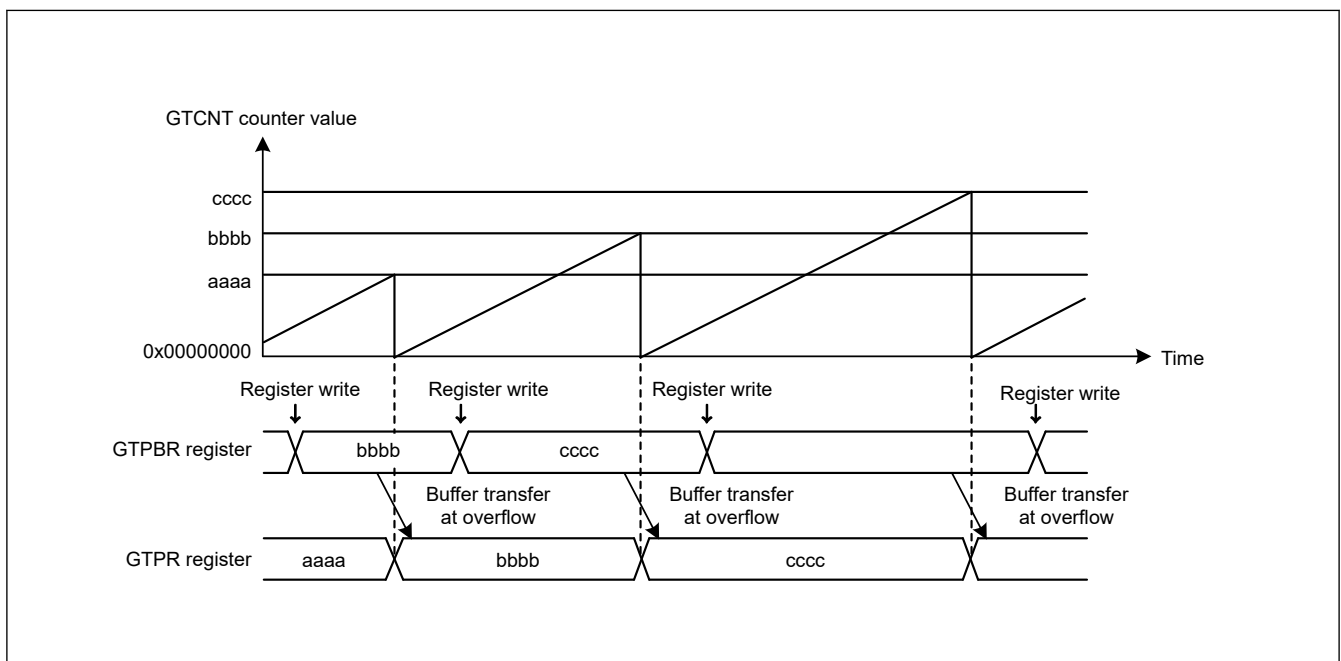


Figure 21.11 Example of GTPR buffer operation with saw waves in up-counting

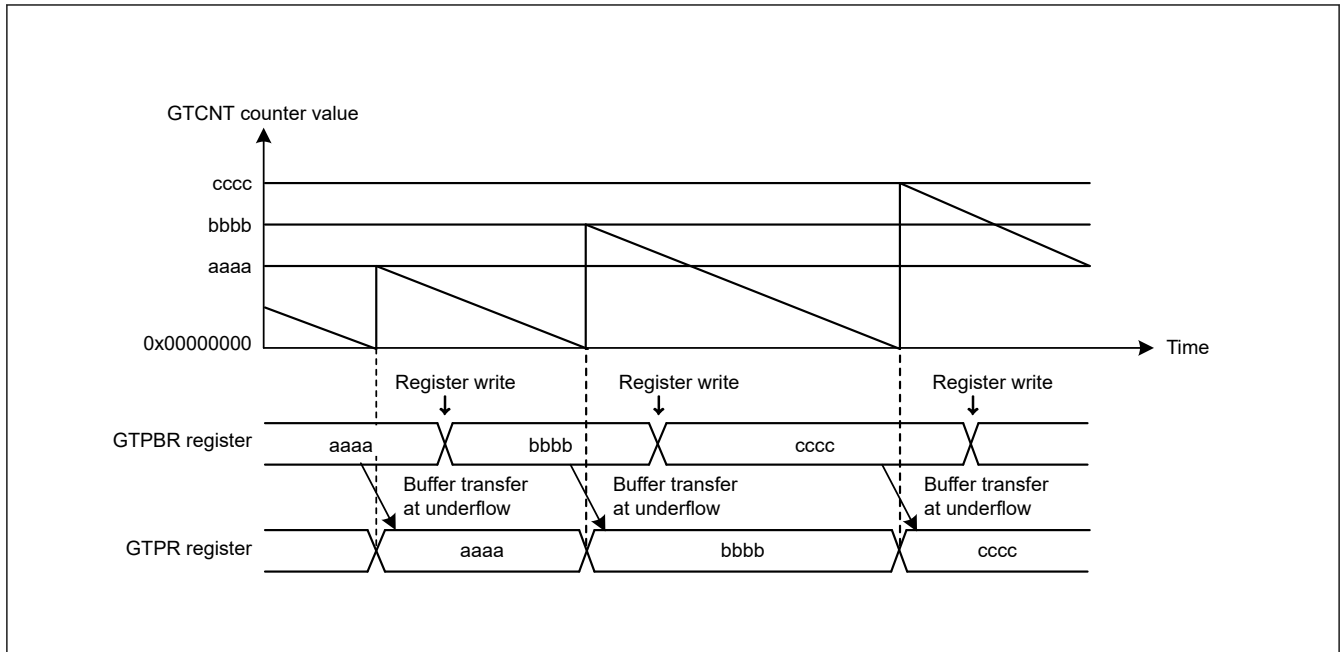


Figure 21.12 Example of GTPR buffer operation with saw waves in down-counting

Table 21.12 Example for setting GTPR register buffer operation

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.11 and Figure 21.12, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.11, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting). In Figure 21.12, after 10b is set in the GTUDDTYC[1:0] bits, 00b is set in the GTUDDTYC[1:0] bits (down-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set buffer operation	Set buffer operation with the GTBER.PR[1:0] bits. In Figure 21.11, Figure 21.12, and section 21.3.2.1. GTPR Register Buffer Operation, PR[1:0] = 01b.
7	Set buffer value	For buffer operation, set a value in one cycle after the current cycle in the GTPBR register.
8	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
9	Set buffer value for each cycle	For buffer operation, set a value in one cycle after the current cycle in the GTPBR register.

### 21.3.2.2 Buffer Operation for GTCCRA and GTCCRB Registers

GTCCRC can function as the GTCCRA buffer register and GTCCRD can function as the GTCCRC buffer register (double-buffer register for GTCCRA). Similarly, GTCCRE can function as the GTCCRB buffer register and GTCCRF can function as the GTCCRE buffer register (double-buffer register for GTCCRB).

To set GTCCRA or GTCCRB to function as a double buffer, set GTBER.CCRA[1:0] or GTBER.CCRB[1:0] to 10b or 11b. For single buffer operation, set 01b. To set GTCCRA or GTCCRB to not function as a buffer, set 00b.

In saw-wave one-shot pulse mode, triangle-wave PWM mode 3, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA [1:0] bits and GTBER.CCRB [1:0] bits.

#### (1) When GTCCRA or GTCCRB Functions as Output Compare Register

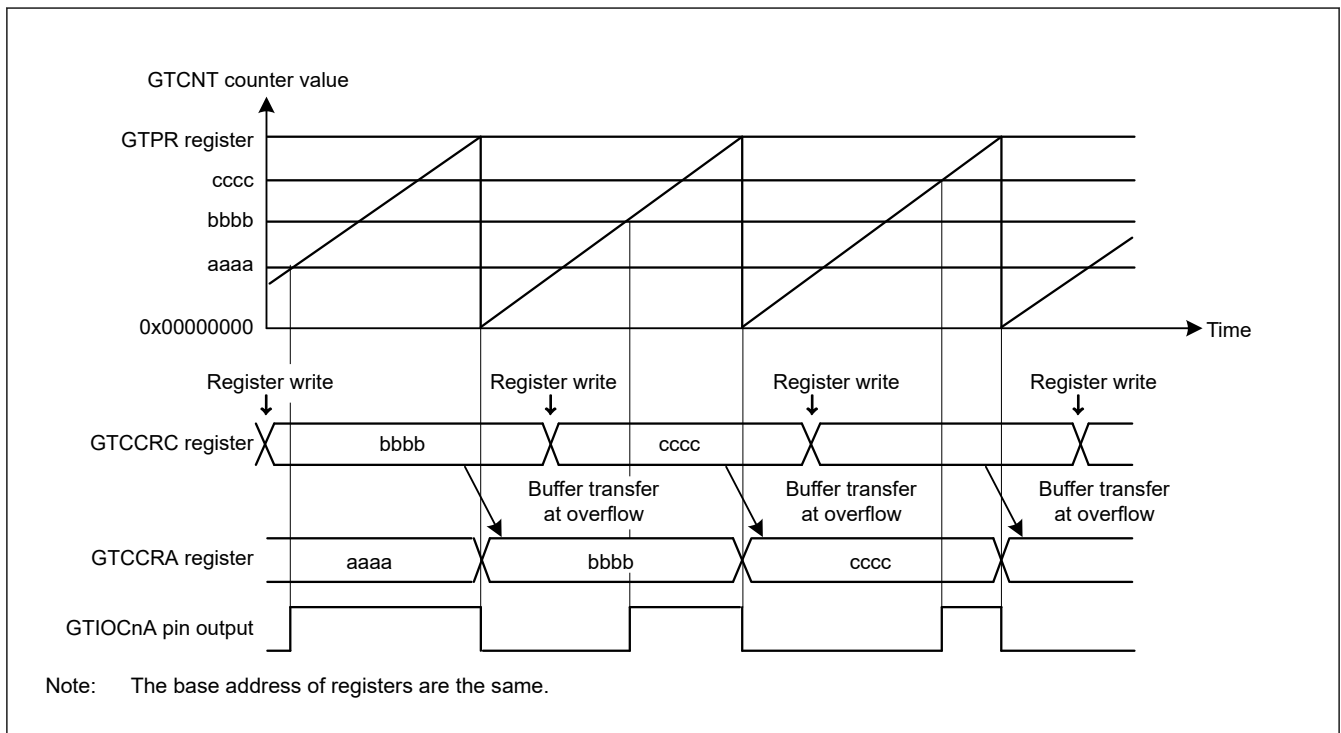
In saw-wave one-shot pulse mode, triangle-wave PWM mode 3, the buffer operations that specific each PWM output operation mode are performed regardless of the setting of GTBER.CCRA [1:0] bits and GTBER.CCRB [1:0] bits. For details, see section 21.3.3. PWM Output Operating Mode. Other than above PWM output operation mode, Buffer transfer occurs in the following situations:



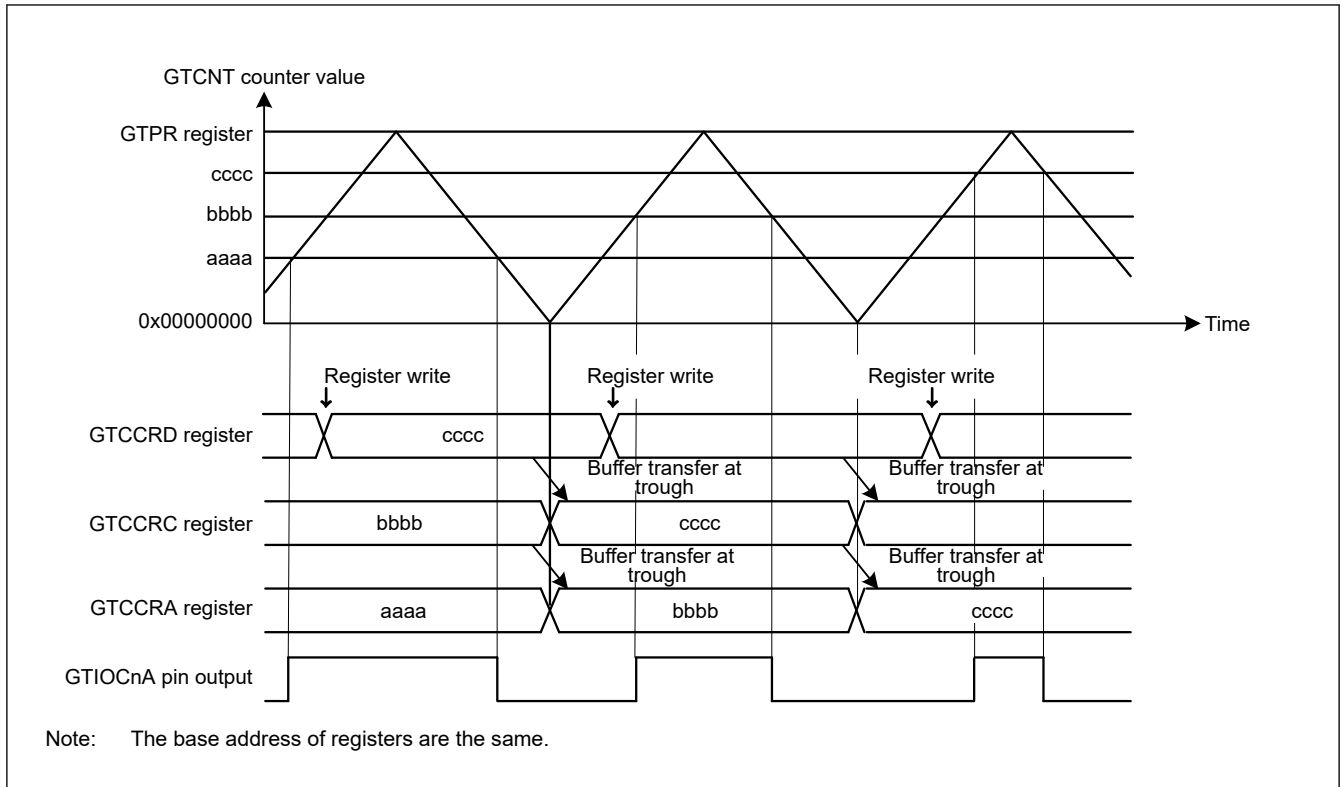
Buffer transfer occurs in the following situations:

- Buffer transfer by overflow or underflow  
Buffer transfer is performed at an overflow (during up-counting) or an underflow (during down-counting) in saw-wave mode or in event count operation. In triangle-wave mode, buffer transfer is performed at a trough (triangle-wave PWM mode 1) or a crest and trough (triangle-wave PWM mode 2).
- Buffer transfer by counter clear  
In saw-wave mode or in event count operation, during counting, buffer transfer (which is the same as an overflow during up-counting or an underflow during down-counting) is performed by the counter clear sources similar to the case shown in [section 21.3.2.1. GTPR Register Buffer Operation](#).  
In triangle-wave mode, buffer transfer is not performed by the counter clear.
- Forcible buffer transfer  
When GTBER.CCRSWT bit is set to 1 while the count operation is stopped, the GTCCRA and the GTCCRB register buffer transfer are performed forcibly in saw-wave mode, in event count operation and in triangle-wave mode.  
Additionally buffer transfer from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B are performed in saw-wave one-shot pulse mode or triangle-wave PWM mode 3.

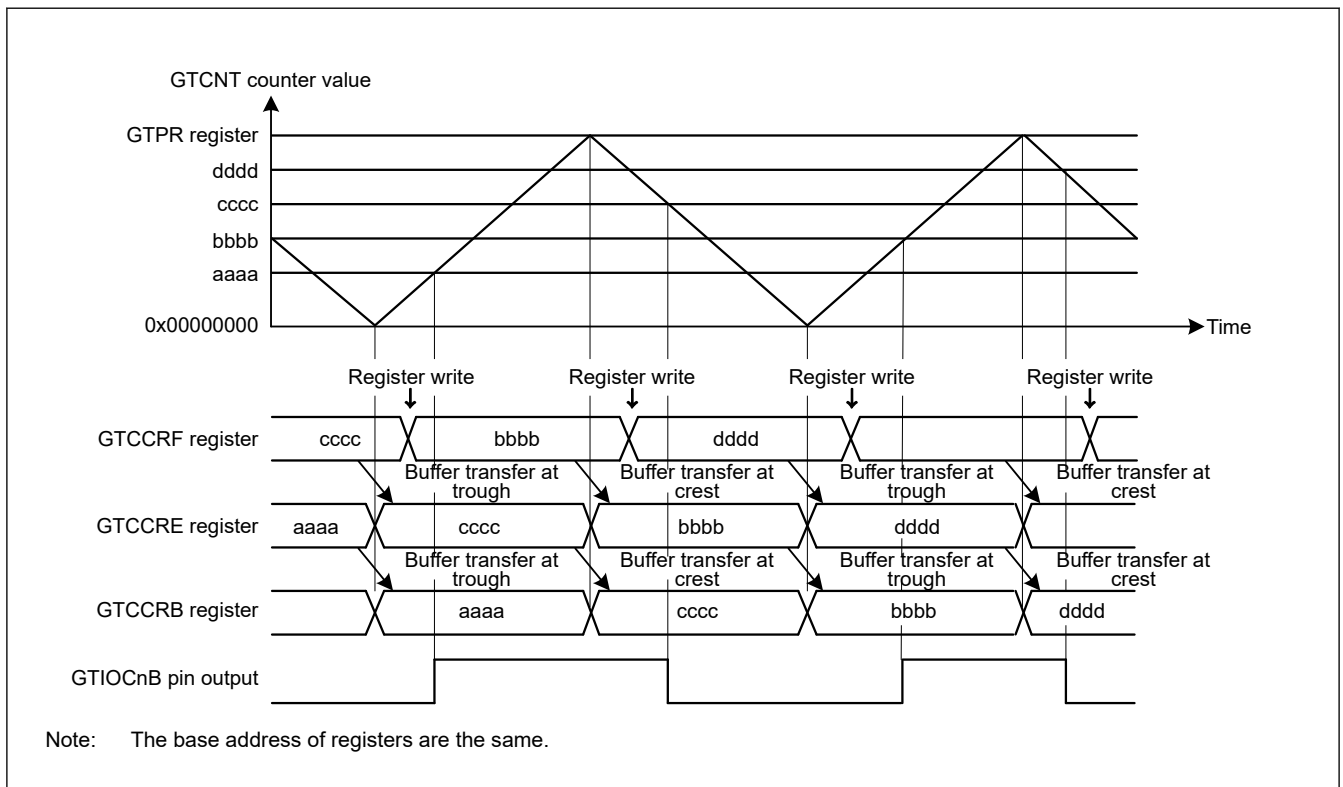
[Figure 21.13](#) to [Figure 21.15](#) show examples of GTCCRA and GTCCRB buffer operation and [Table 21.13](#) shows an example for setting GTCCRA and GTCCRB buffer operation.



**Figure 21.13** Example of GTCCRA and GTCCRB buffer operation with output compare, saw waves in up-counting, high output at GTCCRA compare match, and low output at cycle end



**Figure 21.14** Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at trough, output toggled at GTCCRA compare match, and output retained at cycle end



**Figure 21.15** Example of GTCCRA and GTCCRB double buffer operation with output compare, triangle waves, buffer operation at both troughs and crests, output toggled at GTCCRB compare match, and output retained at cycle end

**Table 21.13 Example for setting GTCCRA and GTCCRB buffer operation for output compare**

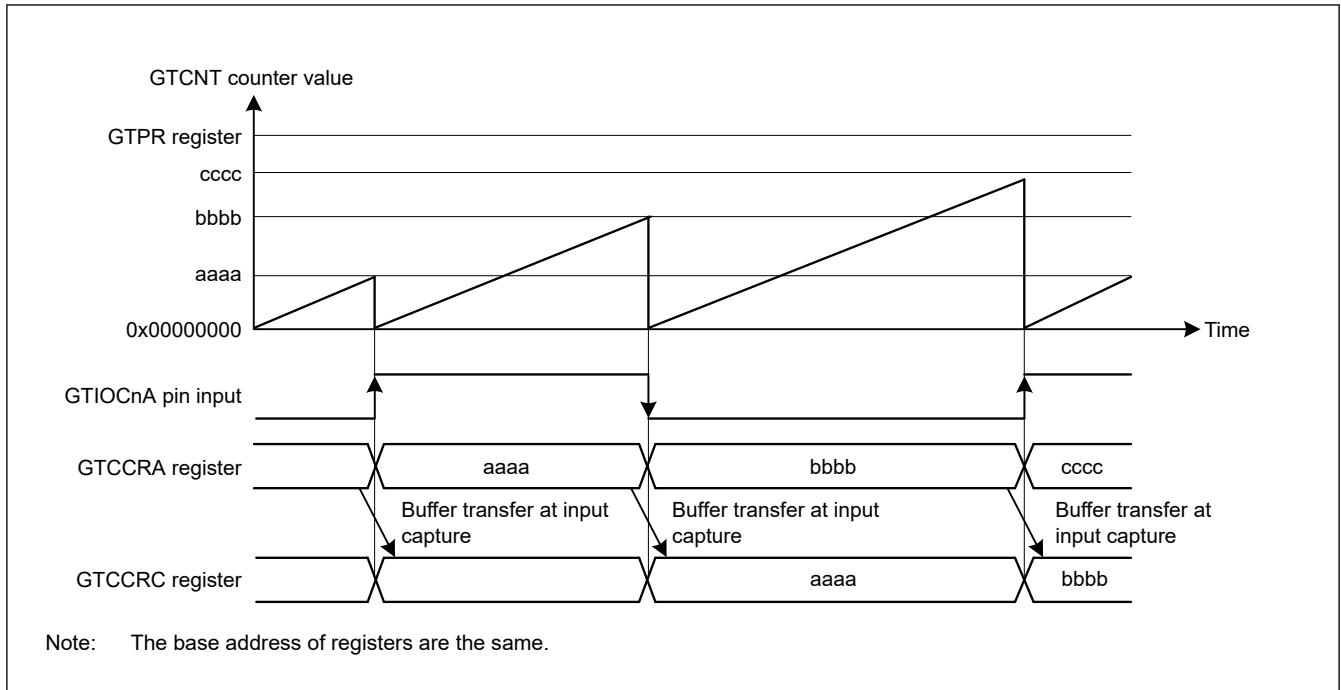
No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In <a href="#">Figure 21.13</a> , 000b (saw-wave PWM mode) is set, in <a href="#">Figure 21.14</a> , 100b (triangle-wave PWM mode 1) is set, and in <a href="#">Figure 21.15</a> , 101b (triangle-wave PWM mode 2) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In <a href="#">Figure 21.13</a> , after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCNm pin function	Set the GTIOCNm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In <a href="#">Figure 21.13</a> , GTIOA[4:0] = 00110b, in <a href="#">Figure 21.14</a> , GTIOA[4:0] = 00011b, and in <a href="#">Figure 21.15</a> , GTIOB[4:0] = 00011b.
7	Enable GTIOCNm pin output	Set to enable the GTIOCNm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTCR register. In <a href="#">Figure 21.13</a> , CCRA[1:0] = 01b, in <a href="#">Figure 21.14</a> , CCRA[1:0] = 1xb, and in <a href="#">Figure 21.15</a> , CCRB[1:0] = 1xb.
9	Set compare match value	Set the GTIOCNm pin transition in the GTCCRA register and the GTIOCNm pin transition in the GTCCRB register.
10	Set buffer value	For buffer operation, set the GTIOCNm and GTIOCNm pins transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCNm and GTIOCNm pins transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCNm and GTIOCNm pins transitions in 1 cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCNm and GTIOCNm pins transitions in 2 cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or 1 cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 5, 10 to 13  
m: A, B

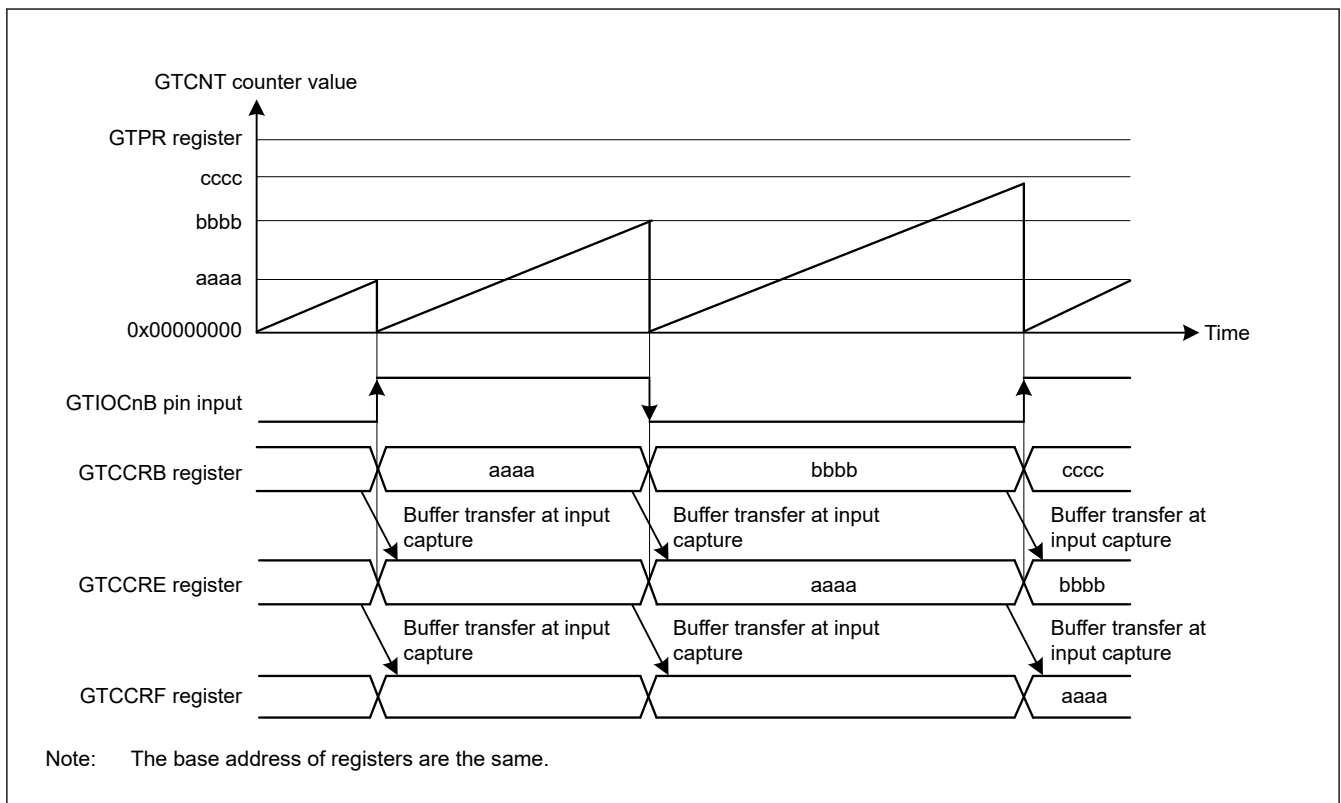
## (2) When GTCCRA or GTCCRB Functions as Input Capture Register

When an input capture is generated, the GTCNT counter value is transferred to GTCCRA and GTCCRB and the stored GTCCRA and GTCCRB register values are transferred to the buffer registers. In input capture operation, the buffer transfer is not performed by the counter clear.

[Figure 21.16](#) and [Figure 21.17](#) show examples of GTCCRA and GTCCRB buffer operation and [Table 21.14](#) shows an example for setting GTCCRA and GTCCRB buffer operation.



**Figure 21.16 Example of GTCCRA and GTCCRB buffer operation with input capture at both edges of GTIOcNA input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOcNA input**



**Figure 21.17 Example of GTCCRA and GTCCRB double buffer operation with input capture at both edges of GTIOcNB input, saw waves in up-counting, and GTCNT counter cleared at both edges of GTIOcNB input**

**Table 21.14 Example for setting GTCCRA and GTCCRB buffer operation for input capture**

No.	Step Name	Description
1	Set operating mode and counter clear sources	Set the operating mode with the GTCR.MD[2:0] bits and count clear source with the GTCSCR register. In <a href="#">Figure 21.16</a> , MD[2:0] = 000b (saw-wave PWM mode) and GTCSCR = 0x00000F00, and in <a href="#">Figure 21.17</a> , MD[2:0] = 000b (saw-wave PWM mode) and GTCSCR = 0x0000F000.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In <a href="#">Figure 21.16</a> , after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Select input capture source	Select input capture source in the GTICASR register and GTICBSR register. In <a href="#">Figure 21.16</a> , GTICASR = 0x00000F00, and in <a href="#">Figure 21.17</a> , GTICBSR = 0x0000F000.
7	Set buffer operation	Set buffer operation with the CCRA and CCRB bits in the GTBER register. In <a href="#">Figure 21.16</a> , CCRA[1:0] = 01b, and in <a href="#">Figure 21.17</a> , CCRB[1:0] = 1xb.
8	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

### 21.3.2.3 Buffer Operation for GTADTRA and GTADTRB Registers

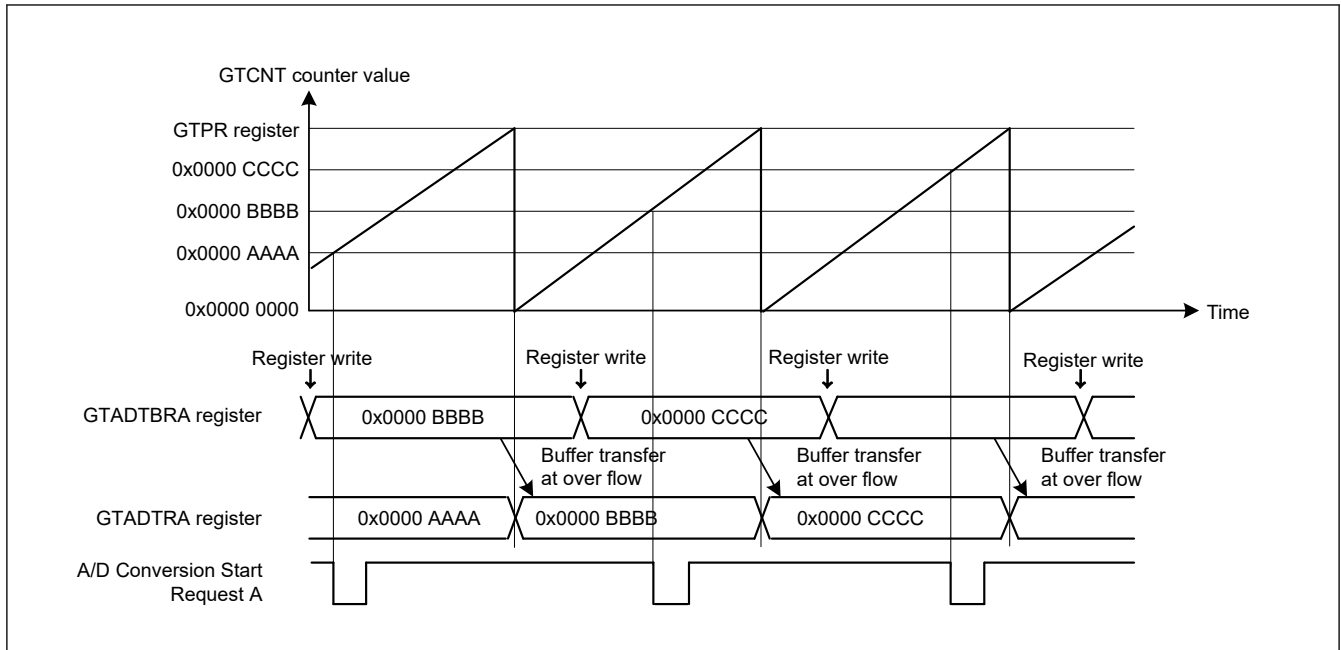
The GTADTBRA register can function as the GTADTRA buffer register and the GTADTDBRA register can function as the GTADTBRA buffer register (double buffer register for the GTADTRA register). Similarly, the GTADTBRB register can function as the GTADTRB buffer register and the GTADTDBRB register can function as the GTADTBRB buffer register (double buffer register for the GTADTRB register).

To set the GTADTRA or GTADTRB register to function as a double buffer, set the GTBER.ADTDA or ADTDB bit to 1. For single buffer operation, set 0. Not to function as buffer, set the GTBER.ADTTA[1:0] or ADTTB[1:0] bits to 00b.

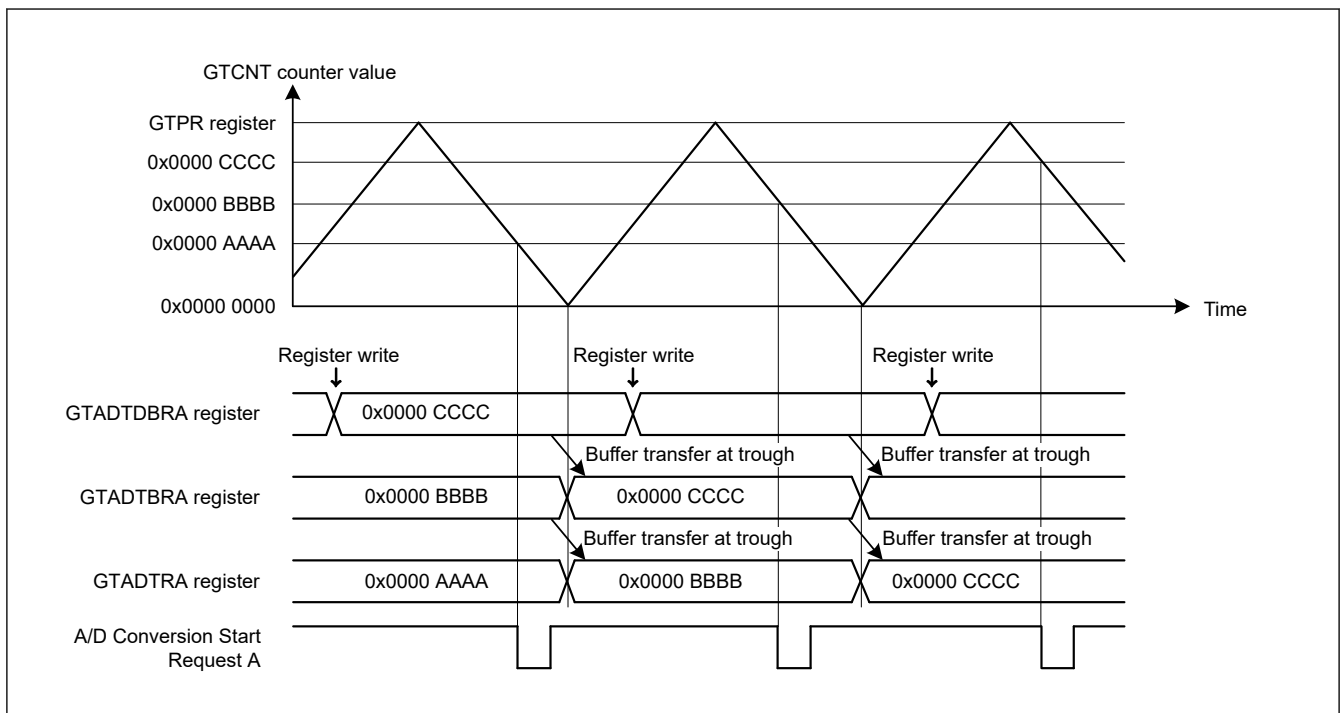
The buffer transfer timing can be set with the ADTTA[1:0] and ADTTB[1:0] bits to an overflow (in up-counting) or an underflow (in down-counting) in saw-wave mode, with ADTTA[1:0] and ADTTB[1:0] bits to 01b for a crest, to 10b for a trough, or to 11b for both crest and trough in triangle-wave mode.

In saw-wave mode, when the ADTTA[1:0] and ADTTB[1:0] bits are set to value other than 00b and in count operation, the buffer transfer, by similar counter clearing sources in [section 21.3.2.1. GTPR Register Buffer Operation](#), is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).

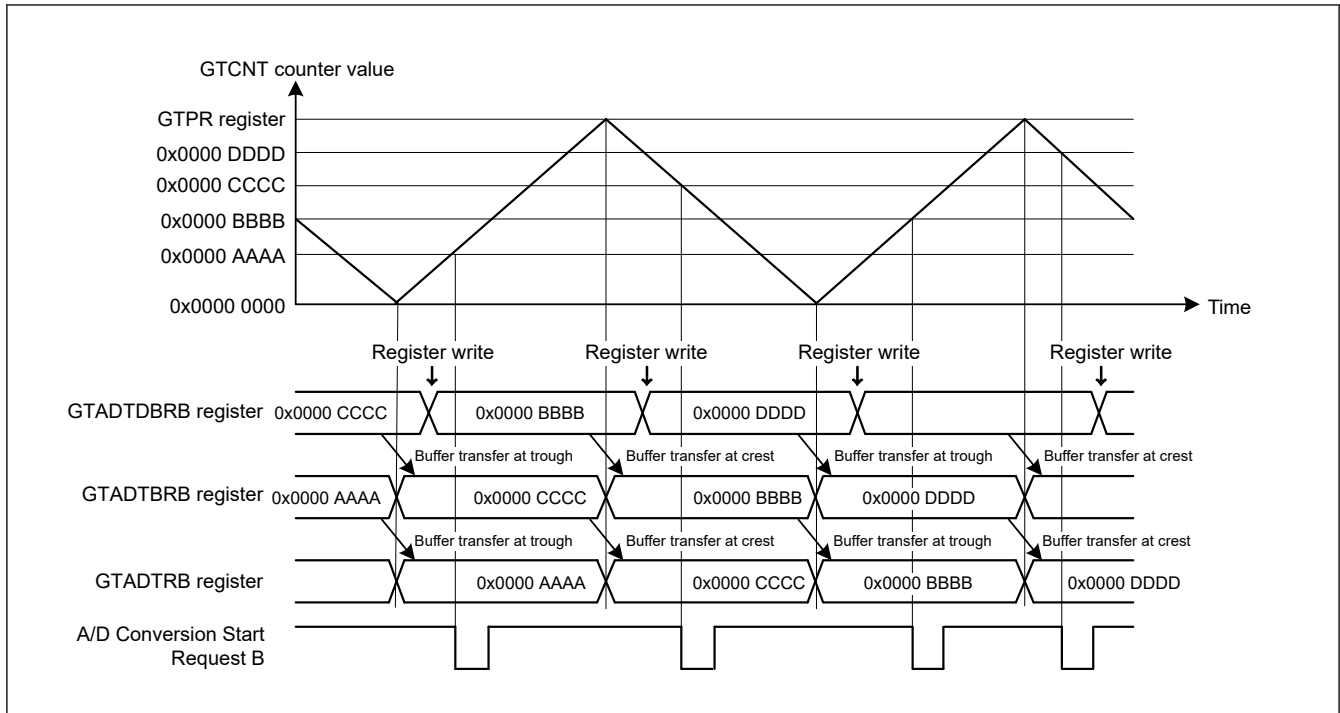
[Figure 21.18](#) to [section 21.3.2.3. Buffer Operation for GTADTRA and GTADTRB Registers](#) show examples of buffer operation of the GTADTRA and GTADTRB registers, and [Table 21.15](#) shows an example for setting buffer operation of the GTADTRA and GTADTRB registers.



**Figure 21.18 Example of Buffer Operation of the GTADTRA and GTADTRB Registers (Saw Waves in Up-Counting, A/D Conversion Start Request Generated by Up-Counting)**



**Figure 21.19 Example of Double Buffer Operation of the GTADTRA and GTADTRB Registers (Triangle Waves, Buffer Transfer at Troughs, A/D Conversion Start Request Generated by Down-Counting)**



**Figure 21.20 Example of Double Buffer Operation of the GTADTRA and GTADTRB Registers (Triangle Waves, Buffer Transfer at Both Troughs and Crests, A/D Conversion Start Request Generated by Both Up- and Down-Counting)**

**Table 21.15 Example for Setting Buffer Operation of the GTADTRA and GTADTRB registers (1 of 2)**

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. (In Figure 21.18 000b (saw-wave PWM mode) is set, in Figure 21.19 and Figure 21.20, 100b, 101b, 110b (triangle-wave PWM mode) is set.)
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. (In Figure 21.18, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).)
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set buffer operation	Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in GTCR register. (In Figure 21.18, ADTTA[1:0] bits = 01b, 10b, or 11b and ADTDA bit = 0, in Figure 21.19, ADTTA[1:0] bits = 10b and ADTDA bit = 1, and in Figure 21.20, ADTTB[1:0] bits = 11b and ADTDB bit = 1.)
7	Set compare match value	Set the A/D conversion start request point in the GTADTRA and GTADTRB registers.
8	Set buffer value	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.
9	Enable A/D conversion start request	Set to enable A/D conversion start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register. (In Figure 21.18, ADTRAUEN bit = 1, in Figure 21.19, ADTRADEN bit = 1, and in Figure 21.20, ADTRBUEN bit = 1 and ADTRBDEN bit = 1.)
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

**Table 21.15 Example for Setting Buffer Operation of the GTADTRA and GTADTRB registers (2 of 2)**

No.	Step Name	Description
11	Set buffer value of each cycle	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers.

### 21.3.3 PWM Output Operating Mode

The GPT can output PWM waveforms to the GTIOcNA or GTIOcNB pin (n = 0 to 5, 10 to 13) by a compare match between the GTCNT counter and GTCCRA or GTCCRB.

By setting GTDTCR and GTDVU, the compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

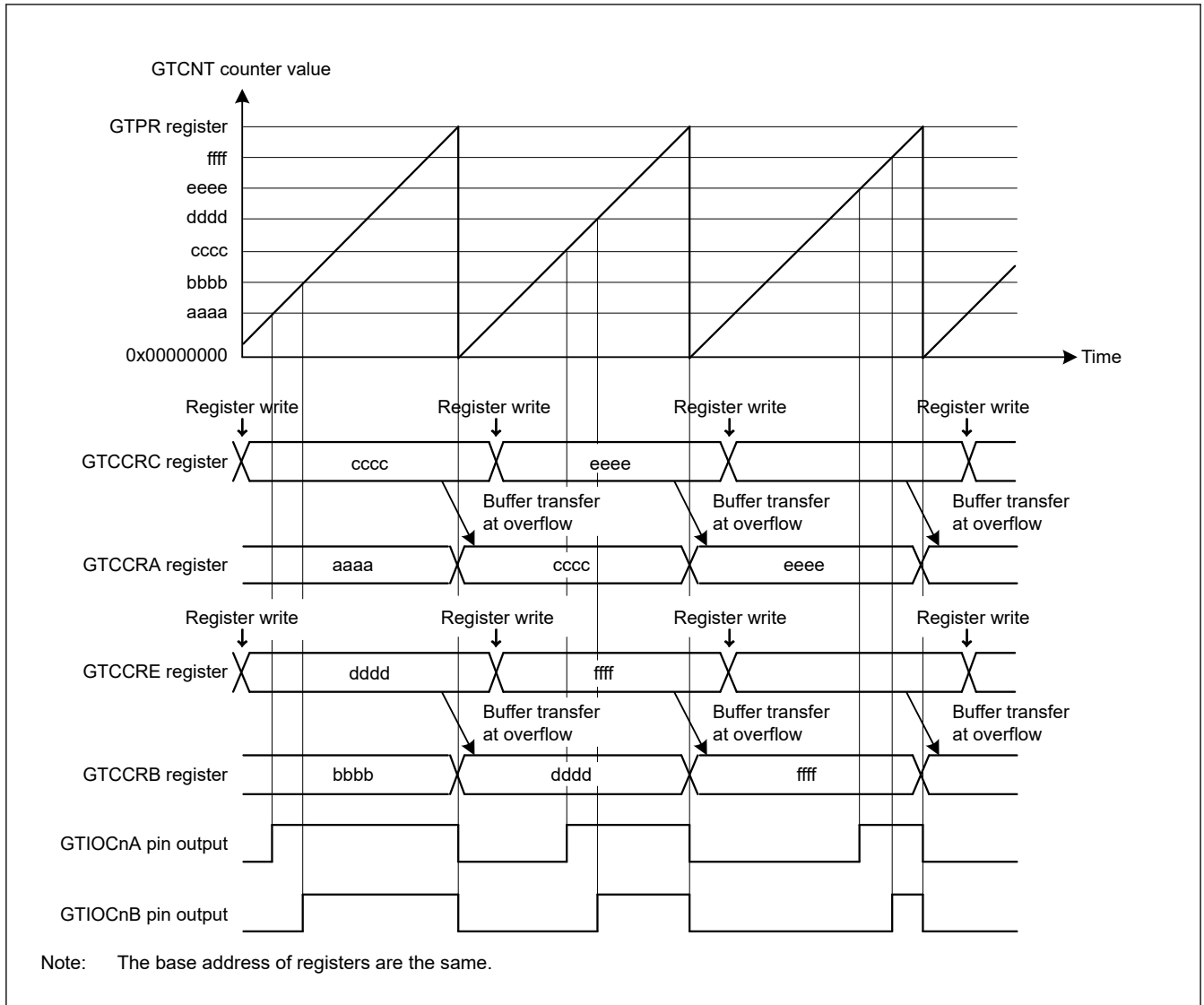
#### 21.3.3.1 Saw-Wave PWM Mode

In saw-wave PWM mode, GTCNT performs saw-wave (half-wave) operation by setting the cycle in GTPR and a PWM waveform is output to the GTIOcNA or GTIOcNB pin (n = 0 to 5, 10 to 13) when a GTCCRA or GTCCRB compare match occurs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

When the timing of the end of cycle and the timing of GTCCRk register compare match are the same time, the output pin performs along the PWM output setting for the end of cycle set by the GTIOR.GTIOx[3:2] bits (x = A, B).

[Figure 21.21](#) shows an example of saw-wave PWM mode operation, and [Table 21.16](#) shows an example for setting saw-wave PWM mode.





**Figure 21.21 Example of saw-wave PWM mode operation with up-counting, buffer operation, high output at GTCCRA/GTCCRB compare match, and low output at cycle end**

**Table 21.16 Example for setting saw-wave PWM mode (1 of 2)**

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.21, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.21, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.21, GTIOA[4:0] = 00110b and GTIOB[4:0] = 00110b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 21.21, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
9	Set compare match value	Set the GTIOCnA pin transition in the GTCCRA register and the GTIOCnB pin transition in the GTCCRB register.

**Table 21.16 Example for setting saw-wave PWM mode (2 of 2)**

No.	Step Name	Description
10	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 5, 10 to 13  
m: A, B

### 21.3.3.2 Saw-Wave One-Shot Pulse Mode

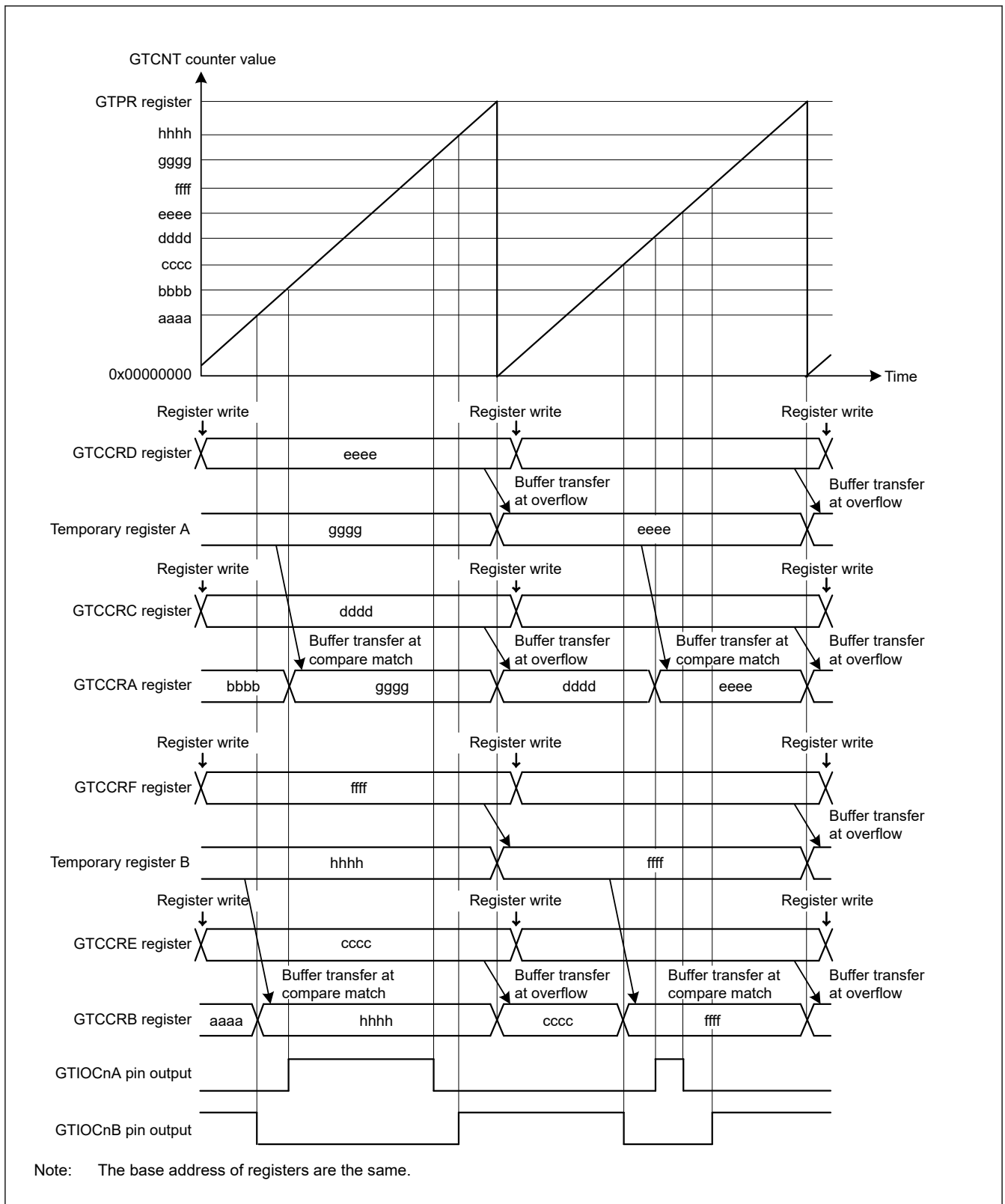
The saw-wave one-shot pulse mode is a mode in which the cycle is set in GTPR, the GTCNT counter performs saw-wave (half-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin ( $n = 0$  to 5, 10 to 13) at a compare match of GTCCRA or GTCCRB with buffer operation fixed.

Buffer operation in saw-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from:

- GTCCRC to GTCCRA at the cycle end
- GTCCRE to GTCCRB at the cycle end
- GTCCRD to temporary register A at the cycle end
- GTCCRF to temporary register B at the cycle end
- Temporary register A to GTCCRA at a GTCCRA compare match
- Temporary register B to GTCCRB at a GTCCRB compare match.

The pin output value can be selected from low output, high output, or toggled output separately for a compare match and the cycle end according to the GTIOR setting. When the GTBER.CCRSWT bit is set to 1 while count operation is stopped, the buffer is transferred forcibly from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B. By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 21.22 shows an example of saw-wave one-shot pulse mode operation, and Table 21.17 shows an example for setting saw-wave one-shot pulse mode.



**Figure 21.22 Example of saw-wave one-shot pulse mode operation with up-counting, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB compare match, and output retained at cycle end**

**Table 21.17 Example setting for saw-wave one-shot pulse mode**

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In <a href="#">Figure 21.22</a> , 001b (saw-wave one-shot pulse mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In <a href="#">Figure 21.22</a> , after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCn pin function	Set the GTIOCn pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In <a href="#">Figure 21.22</a> , GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
7	Enable GTIOCn pin output	Set to enable the GTIOCn pin output with the OAE and OBE bits in the GTIOR register.
8	Set compare match value	Set the GTIOCnA pin transition immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
9	Set forcible buffer transfer	Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly.
10	Set buffer value	For buffer operation, set the GTIOCnA pin transition in one cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.
11	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
12	Set buffer value for each cycle	For buffer operation, set the GTIOCnA pin transition in one cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOCnB pin transition in the GTCCRE and GTCCRF registers.

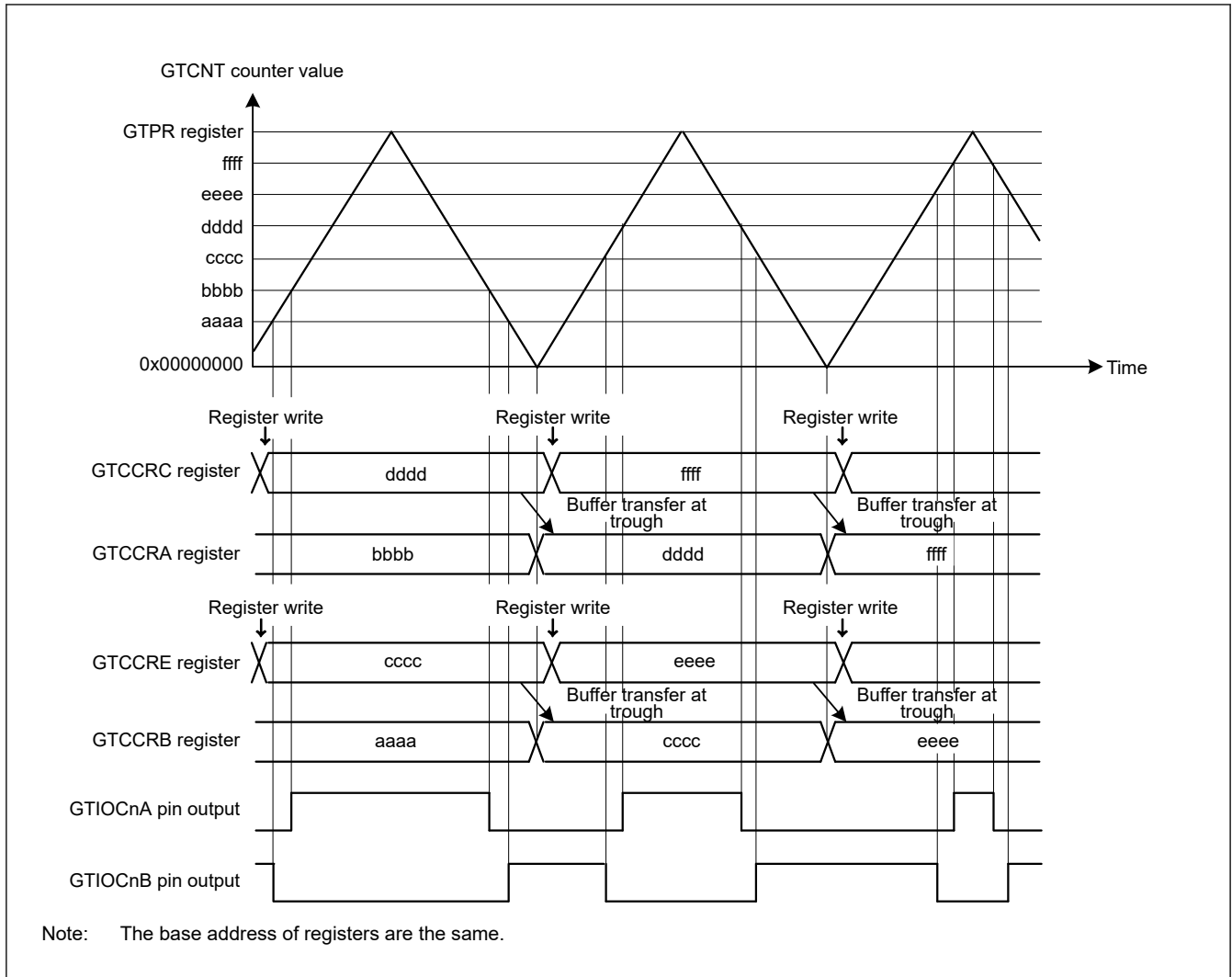
Note: n: 0 to 5, 10 to 13  
m: A, B

### 21.3.3.3 Triangle-Wave PWM Mode 1 (32-Bit Transfer at Trough)

The triangle-wave PWM mode 1 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 5, 10 to 13) when a GTCCRA or GTCCRB compare match occurs. Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggled output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

[Figure 21.23](#) shows an example of a triangle-wave PWM mode 1 operation, and [Table 21.18](#) shows an example for setting a triangle-wave PWM mode 1.



**Figure 21.23 Example of triangle-wave PWM mode 1 operation with buffer operation, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB register compare match, and output retained at cycle end**

**Table 21.18 Example setting for triangle-wave PWM mode 1 (1 of 2)**

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.23, 100b (triangle-wave PWM mode 1) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 21.23, GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
7	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTCR register. In Figure 21.23, CCRA[1:0] = 01b and CCRB[1:0] = 01b.
8	Set compare match value	Set the GTIOCnA and GTIOCnB pins transitions in the GTCCRA and GTCCRB registers, respectively.
9	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

**Table 21.18 Example setting for triangle-wave PWM mode 1 (2 of 2)**

No.	Step Name	Description
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
11	Set buffer value for each cycle	For buffer operation, set the GTIOcNA and GTIOcNB pins transitions in 1 cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOcNA and GTIOcNB pins transitions in 2 cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

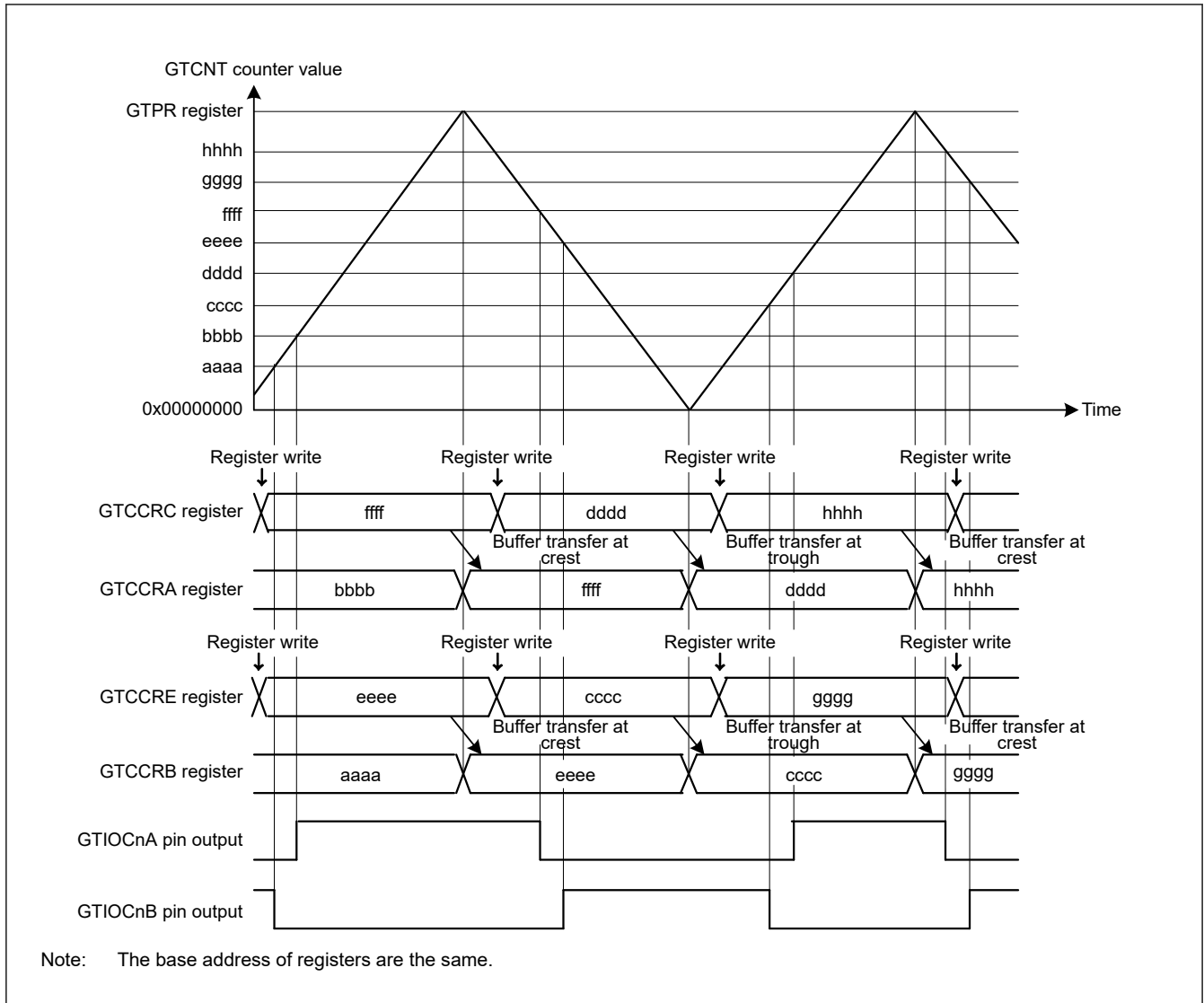
Note: n: 0 to 5, 10 to 13  
m: A, B

### 21.3.3.4 Triangle-Wave PWM Mode 2 (32-Bit Transfer at Crest and Trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2 the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOcNA or GTIOcNB pin (n = 0 to 5, 10 to 13) when a GTCCRA or GTCCRB compare match occurs. The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

[Figure 21.24](#) shows an example of triangle-wave PWM mode 2 operation, and [Table 21.19](#) shows an example for setting triangle-wave PWM mode 2.



**Figure 21.24** Example of triangle-wave PWM mode 2 operation with buffer operation, low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/ GTCCRB compare match, and output retained at cycle end

**Table 21.19** Example for setting triangle-wave PWM mode 2 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In <a href="#">Figure 21.24</a> , 101b (triangle-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In <a href="#">Figure 21.24</a> , GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
7	Set buffer operation	Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTCR register. In <a href="#">Figure 21.24</a> , CCRA[1:0] = 01b and CCRB[1:0] = 01b.
8	Set compare match value	Set the GTIOCnA and GTIOCnB pins transitions in the GTCCRA and GTCCRB registers, respectively.

**Table 21.19 Example for setting triangle-wave PWM mode 2 (2 of 2)**

No.	Step Name	Description
9	Set buffer value	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in the GTCCRD and GTCCRF registers, respectively.
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
11	Set buffer value for each half cycle	For buffer operation, set the GTIOCnA and GTIOCnB pins transitions in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA and GTIOCnB pins transitions in 1 cycle after the current cycle in GTCCRD and GTCCRF registers, respectively.

Note: n: 0 to 5, 10 to 13  
m: A, B

### 21.3.3.5 Triangle-Wave PWM Mode 3 (64-Bit Transfer at Trough)

The triangle-wave PWM mode 3 is a mode in which the cycle is set in GTPR. The GTCNT counter performs triangle-wave (full-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 5, 10 to 13) at a compare match of GTCCRA or GTCCRB with buffer operation fixed. Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from the following:

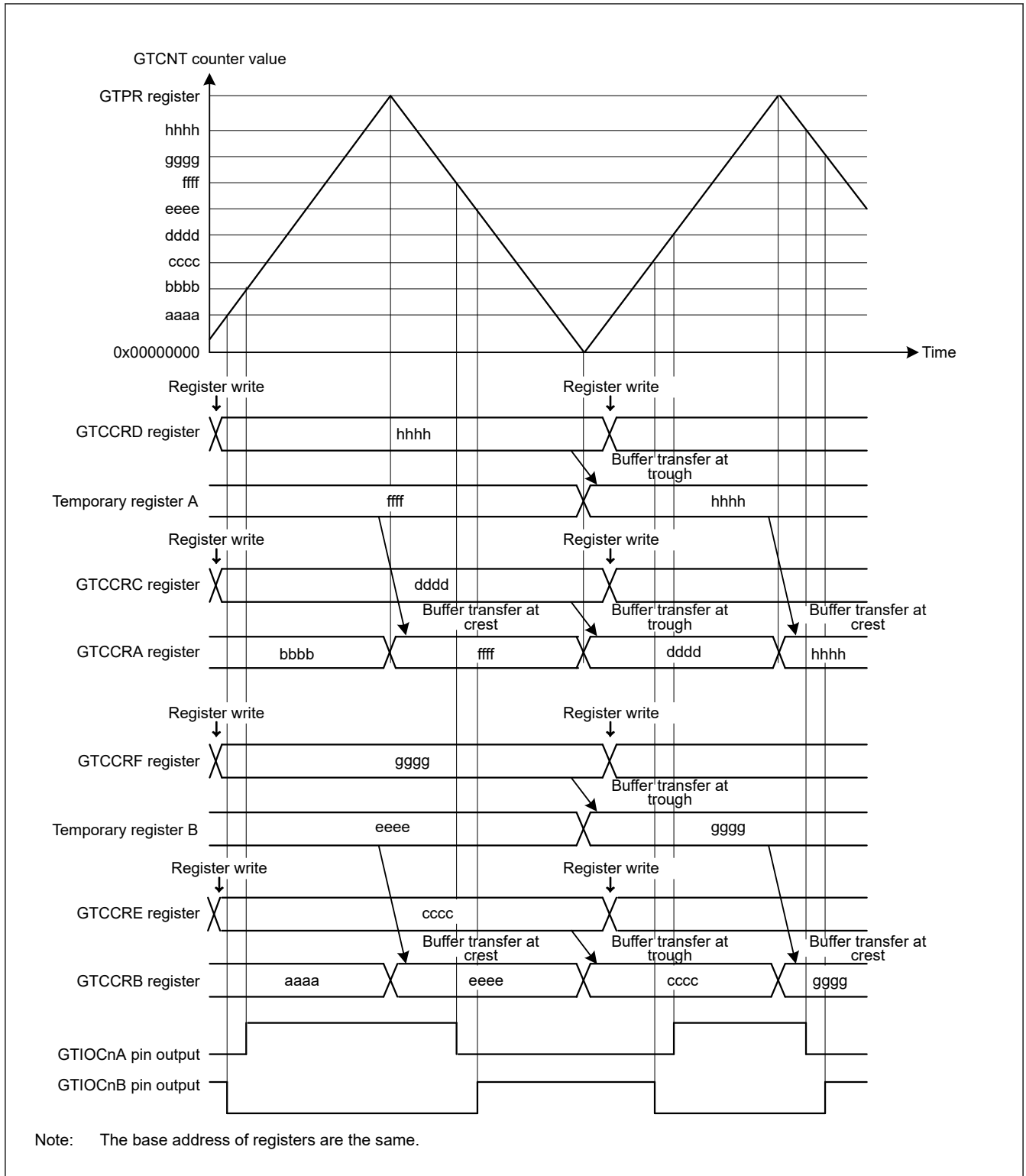
- GTCCRC to GTCCRA at the trough
- GTCCRE to GTCCRB at the trough
- GTCCRD to temporary register A at the trough
- GTCCRF to temporary register B at the trough
- Temporary register A to GTCCRA at the crest
- Temporary register B to GTCCRB at the crest.

The pin output value can be selected from low output, high output, or toggled output separately for a compare match and for the cycle end according to the GTIOR setting.

By setting GTDTCR and GTDVU, a compare match value for a negative-phase waveform with dead time can automatically be set to GTCCRB.

Figure 21.25 shows an example of triangle-wave PWM mode 3 operation, and Table 21.20 shows an example for setting triangle-wave PWM mode 3.





**Figure 21.25 Example of triangle-wave PWM mode 3 operation with low output from the GTIOCnA pin and high output from the GTIOCnB pin at count start, output toggled at GTCCRA/GTCCRB compare match, and output retained at cycle end**

**Table 21.20 Example setting for triangle-wave PWM mode 3 (1 of 2)**

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In <a href="#">Figure 21.25</a> , 110b (triangle-wave PWM mode 3) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.

**Table 21.20 Example setting for triangle-wave PWM mode 3 (2 of 2)**

No.	Step Name	Description
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOcnm pin function	Set the GTIOcnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In <a href="#">Figure 21.25</a> , GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOcnm pin output	Set to enable the GTIOcnm pin output with the OAE and OBE bits in the GTIOR register.
7	Set compare match value	Set the GTIOcnA pin transition immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOcnB pin transition in the GTCCRE and GTCCRF registers.
8	Set forcible buffer transfer	Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly.
9	Set buffer value	Set the GTIOcnA pin transition in 1 cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOcnB pin transition in the GTCCRE and GTCCRF registers.
10	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
11	Set buffer value for each cycle	Set the GTIOcnA pin transition in 1 cycle after the current cycle in the GTCCRC and GTCCRD registers and the GTIOcnB pin transition in the GTCCRE and GTCCRF registers.

Note: n: 0 to 5, 10 to 13  
m: A, B

### 21.3.4 Automatic Dead Time Setting Function

By setting GTDTCR, a compare match value for a negative waveform with dead time obtained by a compare match value for a positive waveform (GTCCRA value) and specified dead time value (GTDVU value) can automatically be set to GTCCRB.

The automatic dead time setting function can be used in saw-wave one-shot pulse mode and all the triangle PWM modes.

Dead time for the changing point of a negative waveform is set in the GTDVU register.

The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. Writing to GTCCRB is prohibited when the automatic dead time setting function is used.

Do not set the dead time that makes the change point of the waveform exceeding the count period. When any dead-time setting which would generate a dead-time error is made, adjust the change points of the positive- and negative-phase waveforms to generate waveforms with secured dead-time as shown in [Table 21.21](#). The adjusted change point of the negative-phase waveform is automatically set in the GTCCRB register. An internal signal is used to judge the change point of the positive-phase waveform, thus the value of the GTCCRA register is not updated by the adjusted value.

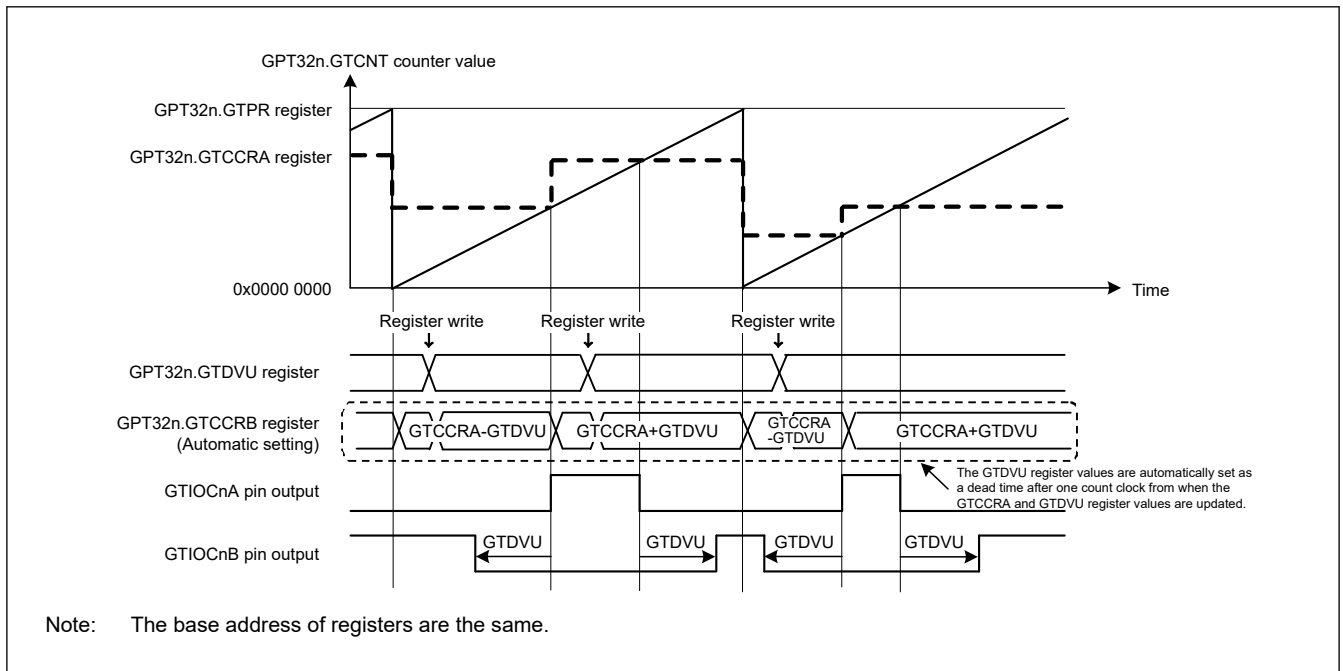
In saw-wave one-shot pulse mode, if the order of the change point becomes inconsistent by adjustment of the waveform change point due to occurrence of dead time errors, or if the change point exceeds the count period even after the adjustment, the complementary relation between the positive- and negative-phases cannot be guaranteed.

Automatic setting for a dead time value to the GTCCRB register is performed at the next count clock after the register value for calculating the automatic setting value is updated. In triangle-wave mode, it also can be done at the next count clock from the current crest.

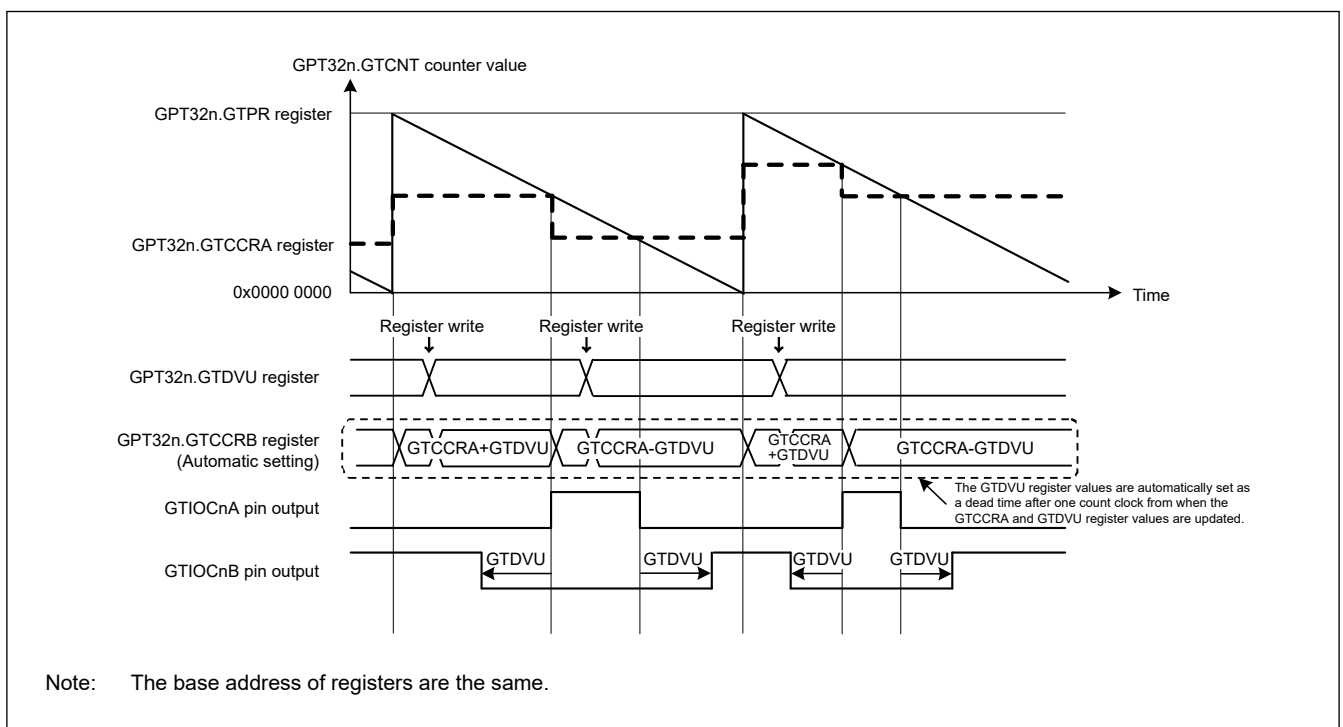
**Table 21.21 Adjustment of the Waveform Change Point When a Dead-Time Error Occurs**

Mode	Count Direction	Period	Condition for Dead Time Error	Change Point of the Positive-Phase Waveform after Adjustment	Change Point of the Negative-Phase Waveform after Adjustment
Sawtooth-wave one-shot pulse mode	Up-counting	First half	$GTCCRA - GTDVU < 0$	GTDVU	0
		Second half	$GTCCRA + GTDVU > GTPR$	$GTPR - GTDVU$	GTPR
	Down-counting	First half	$GTCCRA + GTDVU > GTPR$	$GTPR - GTDVU$	GTPR
		Second half	$GTCCRA - GTDVU < 0$	GTDVU	0
Triangle-wave PWM mode 1/2/3	Up-counting	(First half)	$GTCCRA - GTDVU \leq 0$	$GTDVU + 1$	1
	Down-counting	(Second half)	$GTCCRA - GTDVU < 0$	GTDVU	0

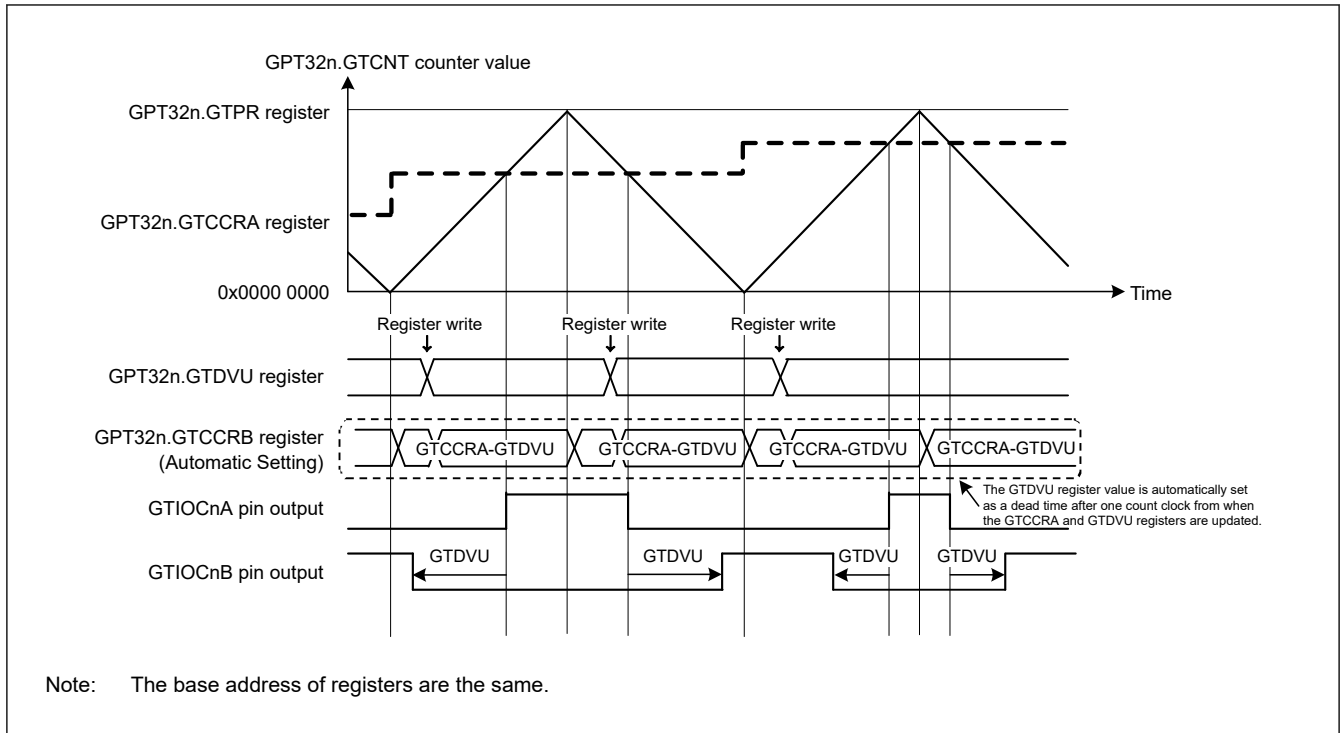
Figure 21.26 to Figure 21.29 show examples of automatic dead time setting function operation. Table 21.22 and Table 21.23 show the setting examples.



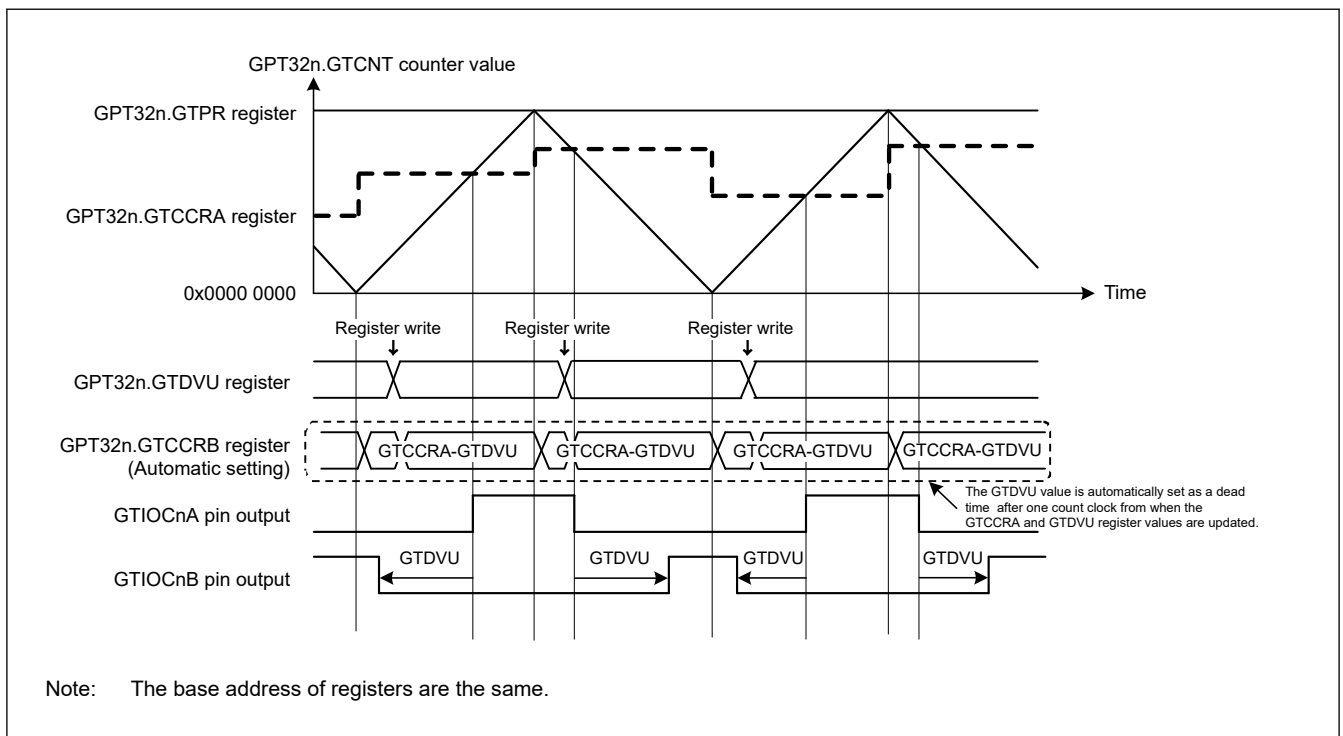
**Figure 21.26 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, up-counting, and active-high**



**Figure 21.27 Example of automatic dead time setting function operation in saw-wave one-shot pulse mode, down-counting, and active-high**



**Figure 21.28** Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 1, and active-high



**Figure 21.29** Example of automatic compare-match value setting function with dead time in triangle-wave PWM mode 2 or 3, and active-high

**Table 21.22** Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3 (1 of 2)

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.26 and Figure 21.27, 001b (saw-wave one-shot pulse mode) is set. In Figure 21.29, 110b (triangle-wave PWM mode 3) is set.

**Table 21.22 Example setting for automatic dead time setting function in saw-wave one-shot pulse mode, and triangle-wave PWM mode 3 (2 of 2)**

No.	Step Name	Description
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In <a href="#">Figure 21.26</a> , 01b is set after 11b is set in the GTUDDTYC[1:0] bits (up count). In <a href="#">Figure 21.27</a> , 00b is set after 10b is set in the GTUDDTYC[1:0] bits (down count).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter.
6	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In <a href="#">Figure 21.26</a> , <a href="#">Figure 21.27</a> , and <a href="#">Figure 21.29</a> , GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
7	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8	Set buffer value for compare match	Set the GTIOCnA pin transition immediately after the count start in the GTCCRC and GTCCRD registers.
9	Set forcible buffer transfer for compare match	Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly to the GTCCRA register.
10	Set buffer value for compare match	Set the GTIOCnA pin transition in 1 cycle after the current cycle in the GTCCRC and GTCCRD registers.
11	Set automatic dead time setting function	Set the GTDTCR.TDE bit to 1 to enable the automatic dead time setting function.
12	Set dead time value	Set the first half dead time value in GTDVU.
13	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
14	Set buffer value for each cycle	Set the GTIOCnA pin transition in 1 cycle after the current cycle in GTCCRC and GTCCRD.

Note: n: 0 to 5, 10 to 13  
m: A, B

**Table 21.23 Example setting for automatic dead time setting function in triangle-wave PWM mode 1 or 2 (1 of 2)**

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In <a href="#">Figure 21.28</a> , 100b (triangle-wave PWM mode 1) is set. In <a href="#">Figure 21.29</a> , 101b (triangle-wave PWM mode 2) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set GTIOCnm pin function	Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In <a href="#">Figure 21.28</a> and <a href="#">Figure 21.29</a> , GTIOA[4:0] = 00011b and GTIOB[4:0] = 10011b.
6	Enable GTIOCnm pin output	Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
7	Set buffer operation for compare match	Set buffer operation with the CCRA[1:0] bits in the GTBER register.
8	Set compare match value	Set the GTIOCnA pin transition in the GTCCRA register.
9	Set buffer value for compare match	For buffer operation, set the GTIOCnA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in the GTCCRC register. For double buffer operation, also set the GTIOCnA pin transition in 2 cycles after the current cycle (in triangle-wave PWM mode 1) or 1 cycle after the current cycle (in triangle-wave PWM mode 2) in the GTCCRD registers.
10	Set automatic dead time setting function	Set the GTDTCR.TDE bit to 1 to enable the automatic dead time setting function.
11	Set dead time value	Set the first half dead time value in GTDVU.
12	Start count operation	Set the GTCR.CST bit to 1 to start count operation.

**Table 21.23 Example setting for automatic dead time setting function in triangle-wave PWM mode 1 or 2 (2 of 2)**

No.	Step Name	Description
13	Set buffer value for each cycle	When the compare match register is used for buffer operation, set the GTIOCnA pin transition in 1 cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRC. When the compare match register is used for double-buffered operation, set the GTIOCnA pin changing point in two cycles after the current cycle (in triangle-wave PWM mode 1) or one cycle after the current cycle (in triangle-wave PWM mode 2) in GTCCRD.

Note: n: 0 to 5, 10 to 13  
m: A, B

### 21.3.5 Count Direction Changing Function

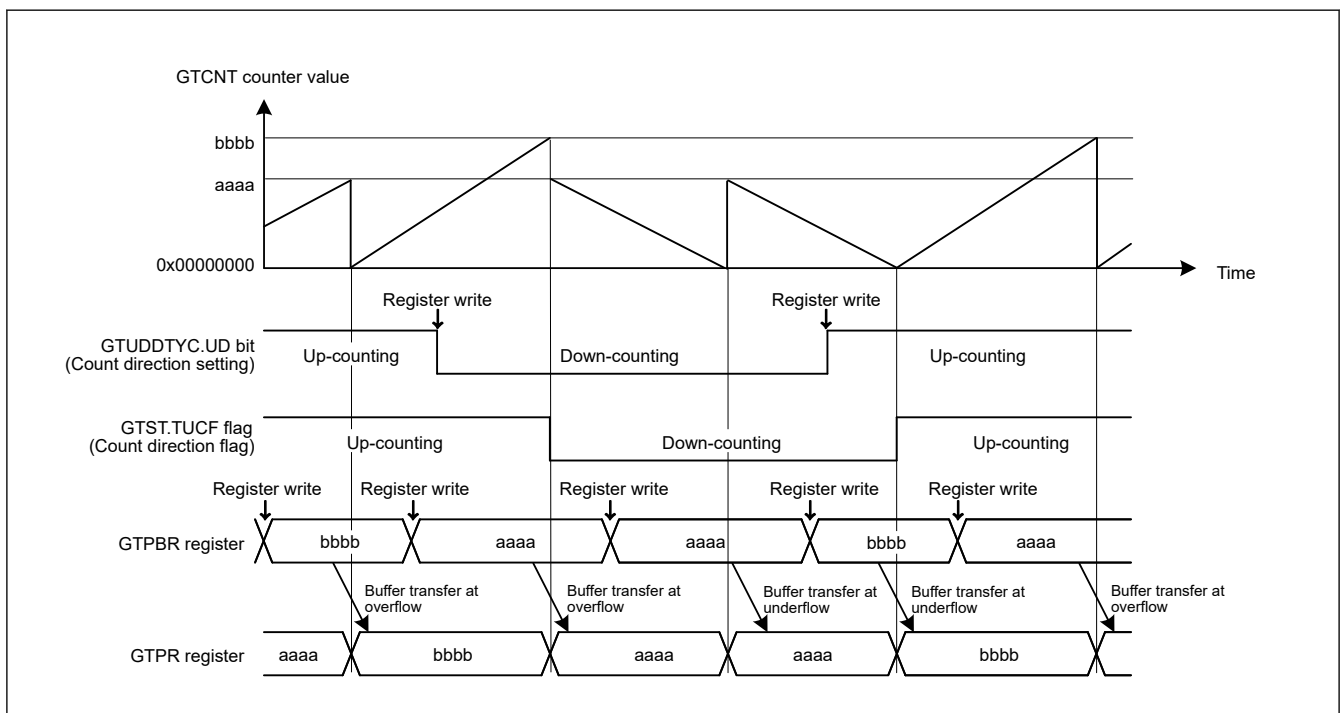
The count direction of the GTCNT counter can be changed by modifying the UD bit in GTUDDTYC.

In saw-wave mode, if the UD bit in GTUDDTYC is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.UD bit is modified while the count operation stops and the GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while the count operation stops, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

In triangle-wave mode, the count direction does not change even though the UD bit in GTUDDTYC is modified during the count operation. Similarly, even though the GTUDDTYC.UD bit is modified while the count operation stops and GTUDDTYC.UDF bit is 0, the GTUDDTYC.UD bit value is not reflected to the count operation. If the GTUDDTYC.UDF bit is set to 1 while the count operation is stopped, the GTUDDTYC.UD bit value at that time is reflected at the start of counting.

If the count direction changes during a saw-wave count operation, the GTPR value after the start of up-counting is reflected in the count cycle during up-counting and the GTPR value after the start of down-counting is reflected in the count cycle during down-counting.

Figure 21.30 shows an example of count direction changing function operation.



**Figure 21.30 Example of a count direction changing function operation during buffer operation**

### 21.3.6 Function of Output Duty 0% and 100%

The output duty of the GTIOCnA pin and the GTIOCnB pin (n = 0 to 5, 10 to 13) are set to 0% or 100% by changing the GTUDDTYC.OADTY bit or GTUDDTYC.OBDTY bit.

In saw-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation, the output duty setting is reflected at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF bit or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an overflow or an underflow. If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF bit or the GTUDDTYC.OBDTYF bit is 1, the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit value at that time is reflected at the start of counting.

In triangle-wave mode, if the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified during the count operation, the output duty setting is reflected an underflow.

If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation is stopped and the GTUDDTYC.OADTYF bit or the GTUDDTYC.OBDTYF bit is 0, the output duty modification is not reflected at the start of counting. The output duty changes at an underflow. If the GTUDDTYC.OADTY bit or the GTUDDTYC.OBDTY bit is modified while the count operation stops and the GTUDDTYC.OADTYF bit or the GTUDDTYC.OBDTYF bit is 1, the output duty modification is reflected at the start of counting.

In performing 0% or 100% duty operation, GPT internally continues to:

- Perform compare match operation
- Set compare match flag
- Output interrupt
- Perform buffer operation.

When the control is changed from 0% or 100% duty setting to compare match, the output value of GTIOCnA pin at cycle end is decided by GTIOR.GTIOA[3:2] and GTUDDTYC.OADTYR. The output value of GTIOCnB pin at cycle end is decided by GTIOR.GTIOB[3:2] and GTUDDTYC.OBDTYR.

When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 01b, the output pins output low at cycle end. When GTIOR.GTIOA[3:2] and GTIOR.GTIOB[3:2] are set to 10b, the output pins output high at cycle end.

GTUDDTYC.OADTYR selects the value that is the object of output retained/toggled at cycle end, when GTIOR.GTIOm[3:2] are set to 00b (output retained at cycle end) or when GTIOR.GTIOm[3:2] are set to 11b (output toggled at cycle end). [Table 21.24](#) shows the values of GTIOCnA and GTIOCnB pin output at cycle end.

**Table 21.24 Output values after releasing 0% or 100% duty setting (m = A, B)**

GTIOR.GTIOm[3:2]	Compare match value at cycle end masked by 0% or 100% duty setting	GTUDDTYC.OmDTYR in duty 0% setting		GTUDDTYC.OmDTYR in duty 100% setting	
		0	1	0	1
00 (output retained at cycle end)	0	0	0	1	0
	1	0	1	1	1
01 (low output at cycle end)	—	0	0	0	0
10 (high output at cycle end)	—	1	1	1	1
11 (output toggled at cycle end)	0	1	1	0	1
	1	1	0	0	0

Figure 21.31 shows an example of output duty 0% and 100% function.

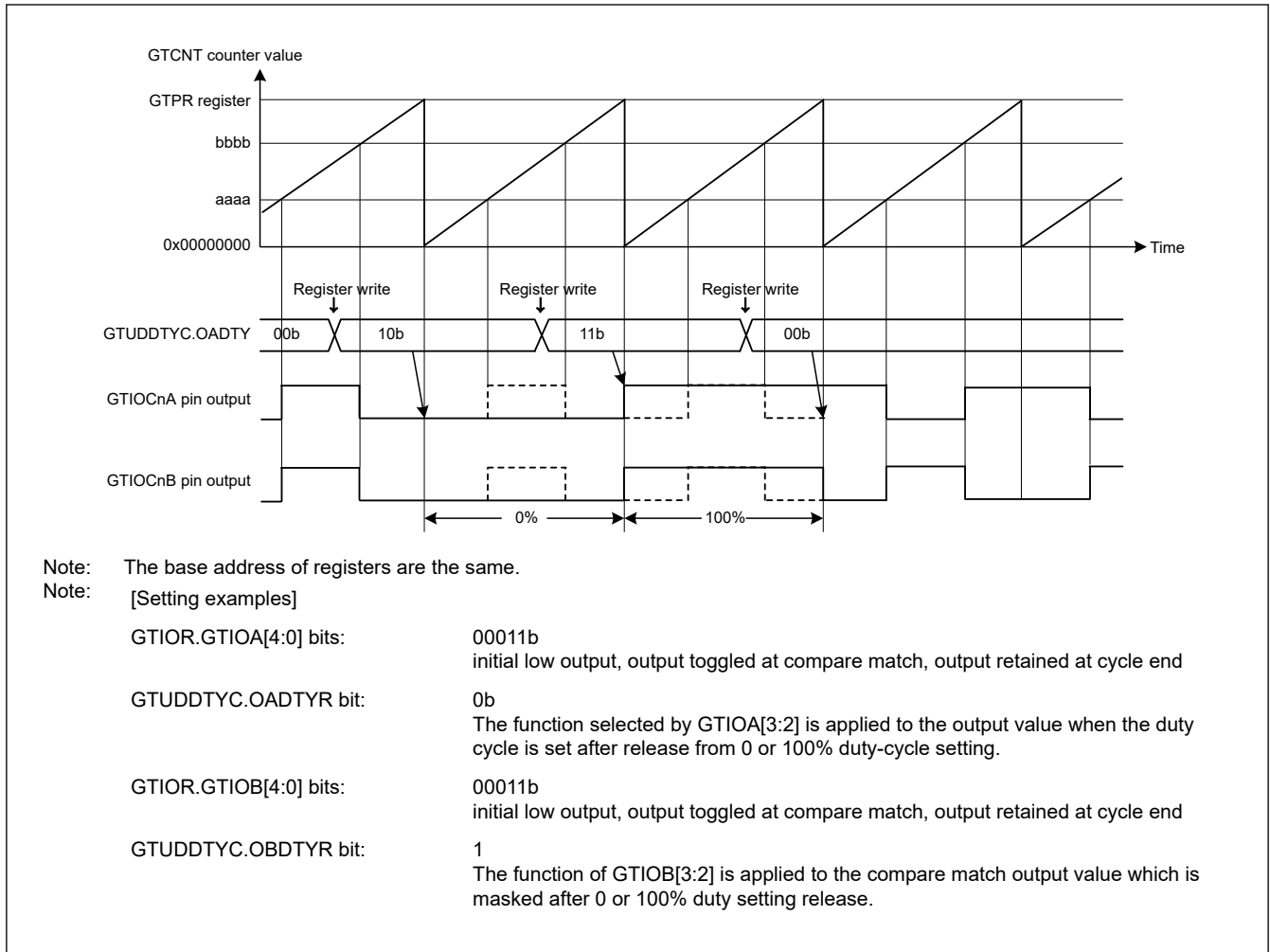


Figure 21.31 Example of output duty 0% and 100% function

### 21.3.7 Hardware Count Start/Count Stop and Clear Operation

The GTCNT counter can be started, stopped, or cleared by the following hardware sources:

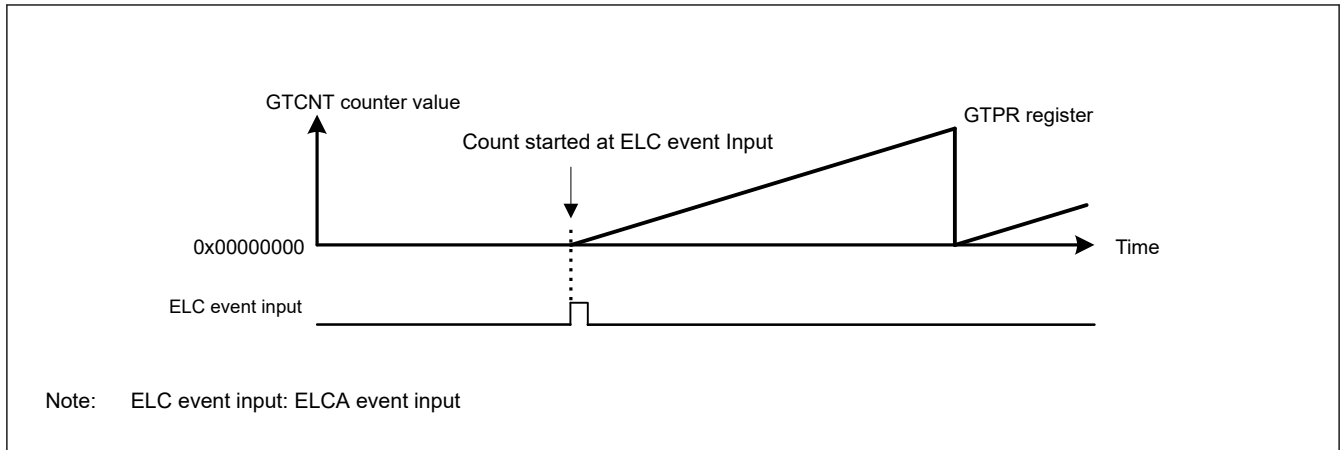
- External trigger input
- ELC event input
- GTIOCnA and GTIOCnB pin input (n = 0 to 5, 10 to 13).

#### 21.3.7.1 Hardware Start Operation

The GTCNT counter can be started by selecting a hardware source using GTSSR.

Figure 21.32 shows an example of a count start operation by a hardware source. Table 21.25 shows the setting example.





**Figure 21.32 Example of count start operation by a hardware source started at the input of the signal from the ELCA event**

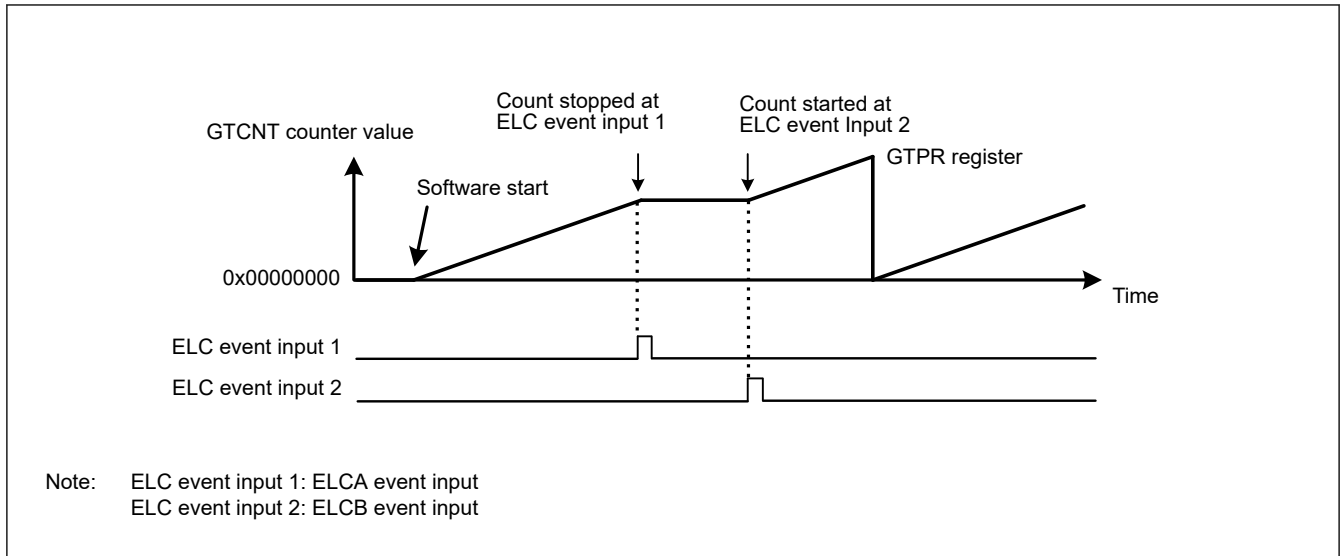
**Table 21.25 Example setting for count start operation by a hardware source**

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In <a href="#">Figure 21.32</a> , 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In <a href="#">Figure 21.32</a> , after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In <a href="#">Figure 21.32</a> , 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation in the GTSSR register. In <a href="#">Figure 21.32</a> , GTSSR.SSELCA = 1
7	Set hardware source operation	Set operation of the hardware source selected by the GTSSR register and start counting. In <a href="#">Figure 21.32</a> , the ELCA event input operation is set.

### 21.3.7.2 Hardware Stop Operation

The GTCNT counter can be stopped by selecting a hardware source using GTPSR.

[Figure 21.33](#) shows an example of a count stop operation by a hardware source. [Table 21.26](#) shows the setting example. In this example, the count operation stops at the ELCA event input and restarts at the ELCB event input.

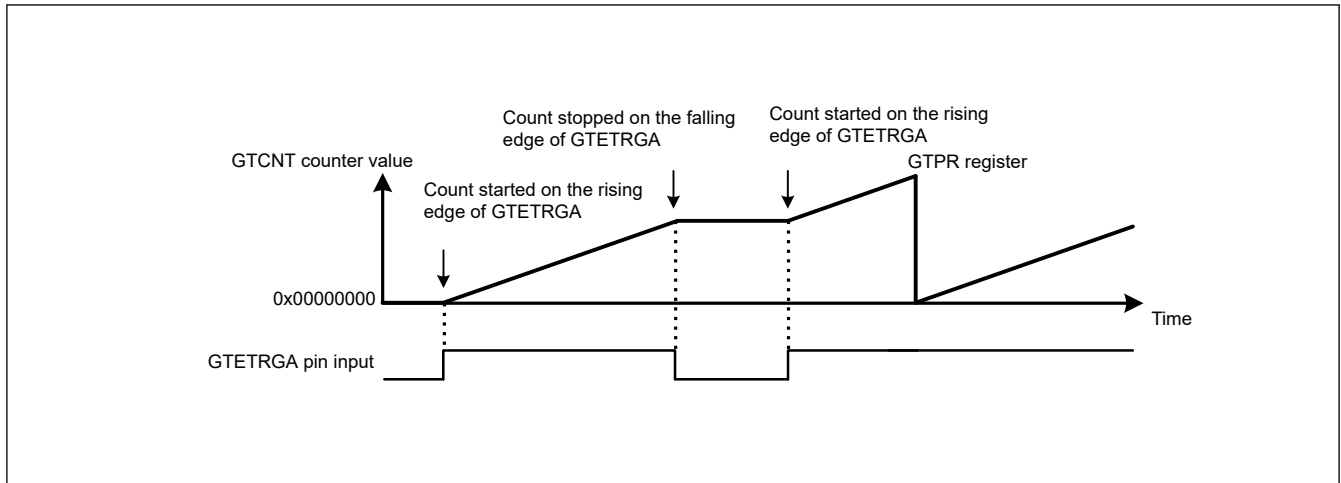


**Figure 21.33 Example of count stop operation by hardware source started by software, stopped at ELCA input, and restarted at ELCB input**

**Table 21.26 Example setting for count stop operation by a hardware source**

No.	Step Name	Description
1	Set operating mode	Set the operating mode with GTCR.MD[2:0] bits. In <a href="#">Figure 21.33</a> , 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In <a href="#">Figure 21.33</a> , after 11b is set in GTUDDTYC[1:0], 01b is set in GTUDDTYC[1:0] (up-counting).
3	Select count clock	Select the count clock with GTCR.TPCS[3:0].
4	Set cycle	Set the cycle in GTPR.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In <a href="#">Figure 21.33</a> , 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation in GTSSR register, and wait for count start by the hardware source. In <a href="#">Figure 21.33</a> , GTSSR.SSELCB = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation in GTPSR register and wait for count stop by the hardware source. In <a href="#">Figure 21.33</a> , GTPSR.PSELCA = 1.
8	Set hardware source operation	Set operation of the hardware source selected in GTSSR register or GTPSR register, and start or stop counting. In <a href="#">Figure 21.33</a> , ELCA input operation and ELCB input operation are set.

[Figure 21.34](#) shows an example of a count start/stop operation by a hardware source. [Table 21.27](#) shows the setting example. In this example, the counter operates during the high-level periods of the external trigger input GTETRGA.



**Figure 21.34** Example of count start/stop operation by a hardware source started on the rising edge of GTETRGA pin input, and stopped on the falling edge of GTETRGA pin input

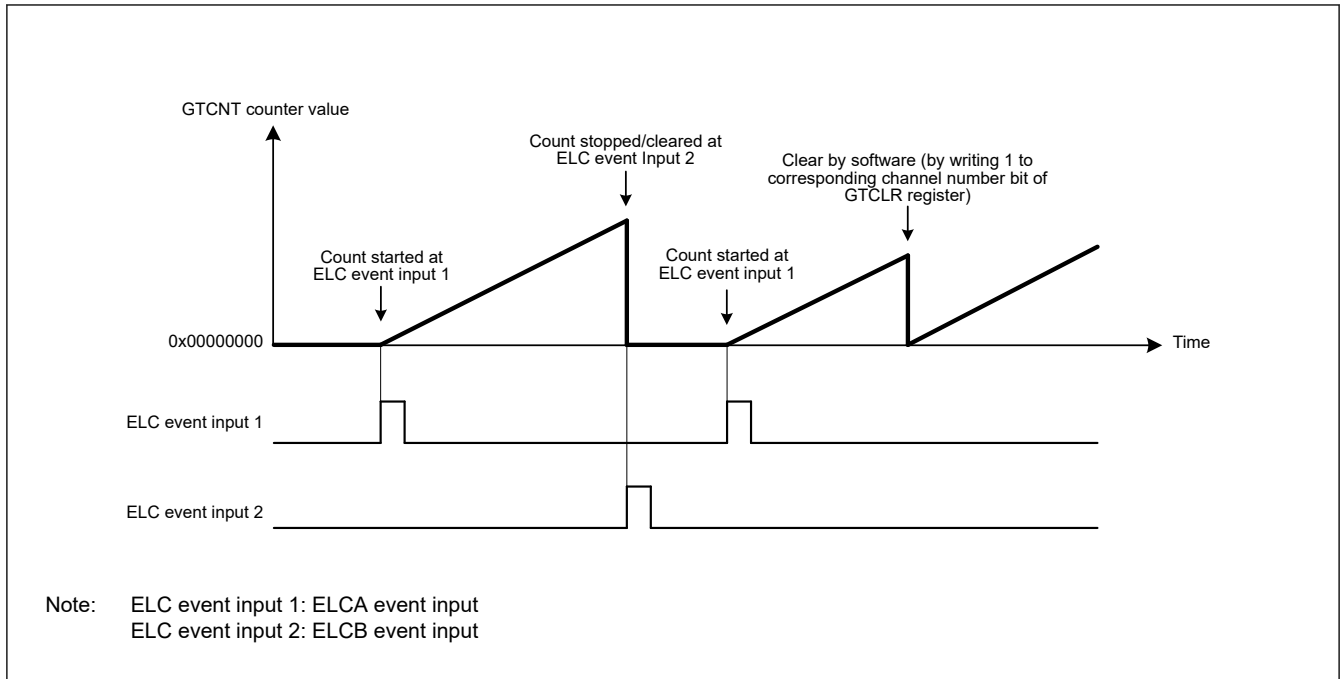
**Table 21.27** Example setting for count start/stop operation by a hardware source

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.34, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.34, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.34, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation with the GTSSR register, and wait for count start by the hardware source. In Figure 21.34, GTSSR.SSGTRGAR = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation with the GTPSR register, and wait for count stop by the hardware source. In Figure 21.34, GTPSR.PSGTRGAF = 1.
8	Set hardware source operation	Set operation of the hardware source selected in the GTSSR register or GTPSR register and start or stop counting. In Figure 21.34, the GTETRGA pin operation is set.

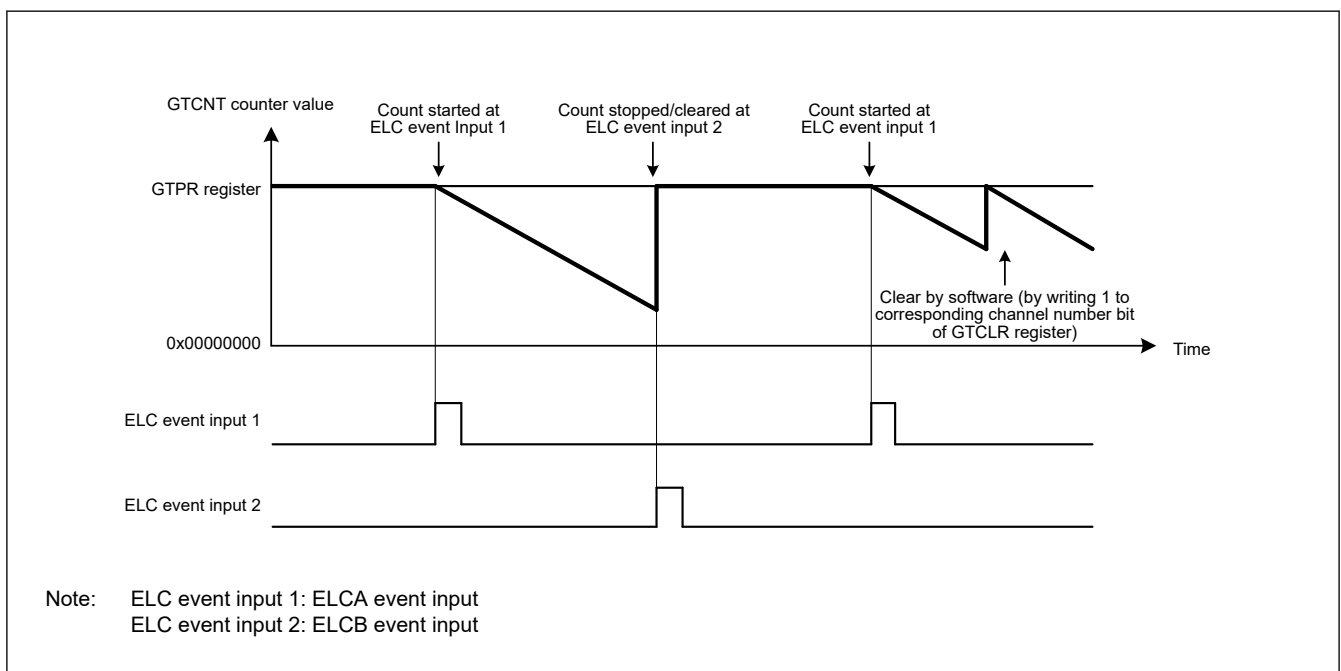
### 21.3.7.3 Hardware Clear Operation

The GTCNT counter can be cleared by selecting a hardware source using GTCSR. The GPT<sub>n</sub>\_OVF/GPT<sub>n</sub>\_UDF (n = 0 to 5, 10 to 13) interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 21.35 and Figure 21.36 show examples of the GTCNT counter clearing operation by a hardware source. Table 21.28 shows the setting example. In this example, the GTCNT counter starts at the ELCA input, and the counter stops and clears at the ELCB input.



**Figure 21.35** Examples of count clearing operation by hardware source in saw wave up-counting, started at ELCA input, and stopped/cleared at ELCB input



**Figure 21.36** Examples of count clearing operation by hardware source in saw wave down-counting, started at ELCA input, and stopped/cleared at ELCB input

**Table 21.28** Example setting for count clearing operation by a hardware source (1 of 2)

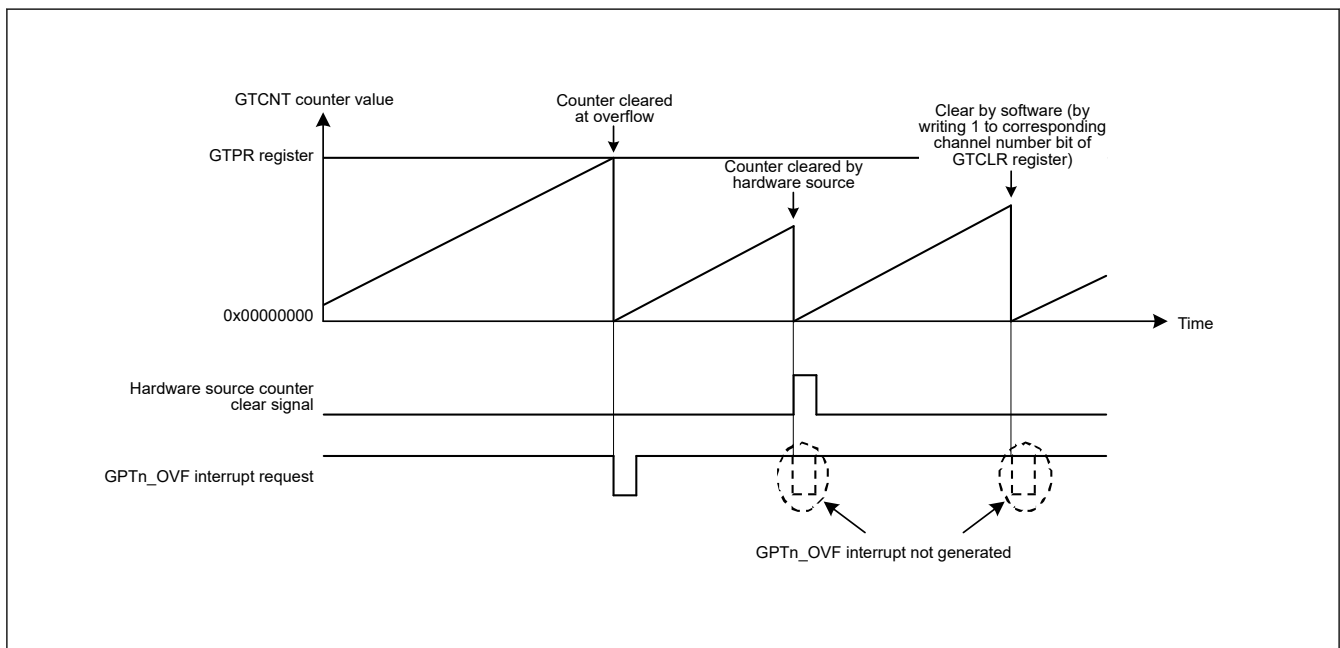
No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In <a href="#">Figure 21.35</a> and <a href="#">Figure 21.36</a> , 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In <a href="#">Figure 21.35</a> , after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting). In <a href="#">Figure 21.36</a> , after 10b is set in the GTUDDTYC[1:0] bits, 00b is set in the GTUDDTYC[1:0] bits (down-counting).

**Table 21.28 Example setting for count clearing operation by a hardware source (2 of 2)**

No.	Step Name	Description
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.35, 0x00000000 is set. In Figure 21.36, the GTPR register value is set.
6	Set hardware count start	Select a hardware source for starting count operation in the GTSSR register, and wait for count start by the hardware source. In Figure 21.35 and Figure 21.36, GTSSR.SSELCA = 1.
7	Set hardware count stop	Select a hardware source for stopping count operation in the GTPSR register, and wait for count stop by the hardware source. In Figure 21.35 and Figure 21.36, GTPSR.PSELCB = 1.
8	Set hardware count clear	Select a hardware source for clearing count operation in the GTCSR register, and wait for count clear by the hardware source. In Figure 21.35 and Figure 21.36, GTCSR.CSELCB = 1.
9	Set hardware source operation	Set operation of the hardware source selected in the GTSSR register, GTPSR register or GTCSR register and start, stop or clear counting. In Figure 21.35 and Figure 21.36, the ELCA input and ELCB input are set.

The GPTn\_OVF/GPTn\_UDF (n = 0 to 5, 10 to 13) interrupt (overflow/underflow interrupt) is not generated when the counter is cleared by a hardware source or by software.

Figure 21.37 shows the relationship between the counter clearing by a hardware source and the GPTn\_OVF (n = 0 to 5, 10 to 13) interrupt.



**Figure 21.37 Relationship between counter clearing by hardware source and GPTn\_OVF (n = 0 to 5, 10 to 13) interrupt**

### 21.3.8 Synchronized Operation

Synchronized operation on channels such as a synchronized start, stop, and clear operation can be performed.

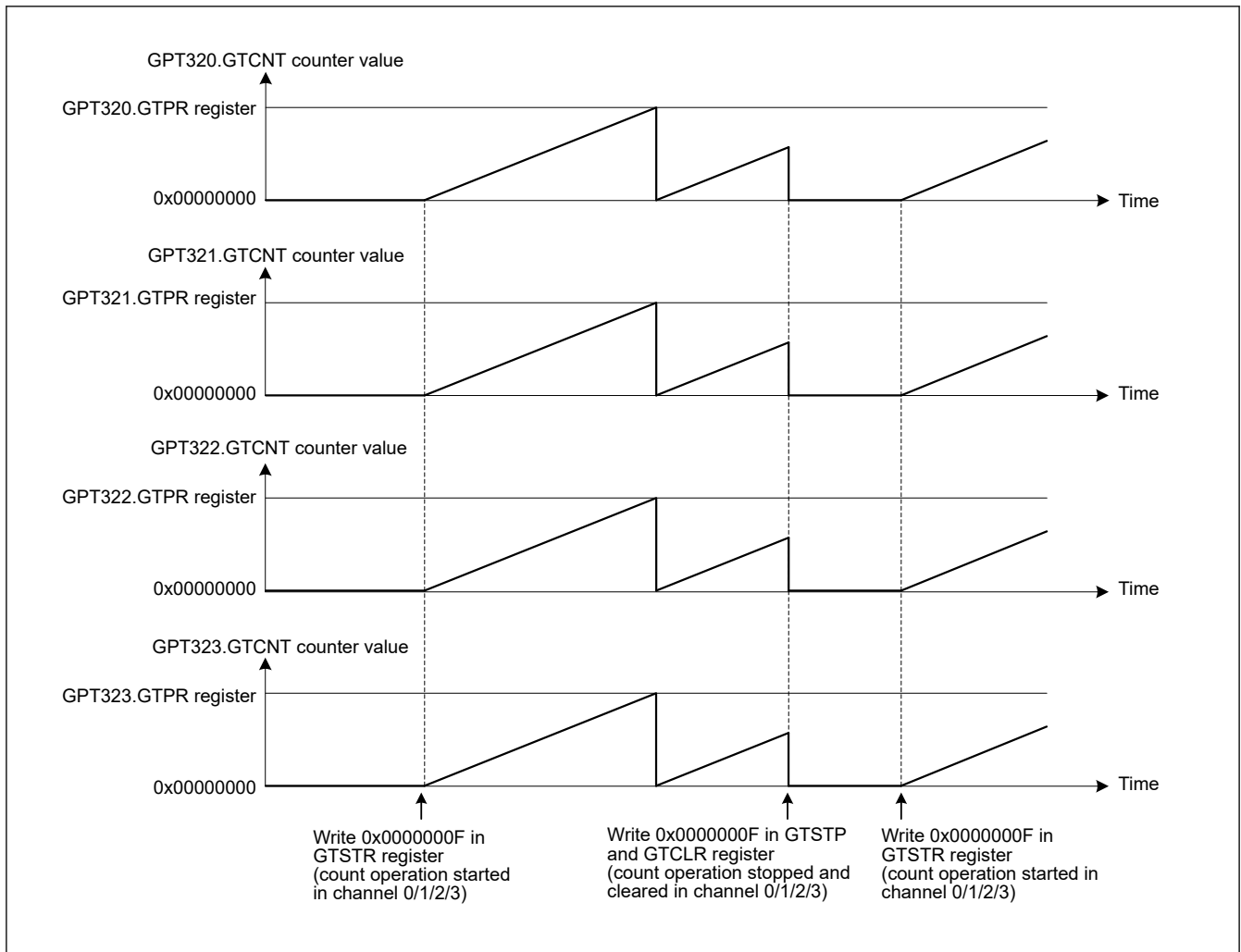
#### 21.3.8.1 Synchronized Operation by Software

The GTCNT counters can be started, stopped, and cleared on multiple channels by setting the associated GTSTR, GTSTP, or GTCLR bits simultaneously to 1.

Count start with a phase difference is possible by setting the initial value in the GTCNT counter and setting the associated GTSTR bits simultaneously to 1.

Because the clock of count operation is selected by GTCR.TPCS[3:0] bits in respective channels, if the clock period of each channel that performs synchronous operation (count start/stop/clear) is different from others, the synchronous operation timings of every channels are not exact same.

Figure 21.38 shows an example of a simultaneous start, stop, and clear by software. Figure 21.39 shows an example of phase start operation by software. Figure 21.40, Figure 21.41, Figure 21.42 show an example of simultaneous start/stop/clearing with different count period.



**Figure 21.38 Example of a simultaneous start, stop, and clear by software with the same count cycle (GTPR register value)**

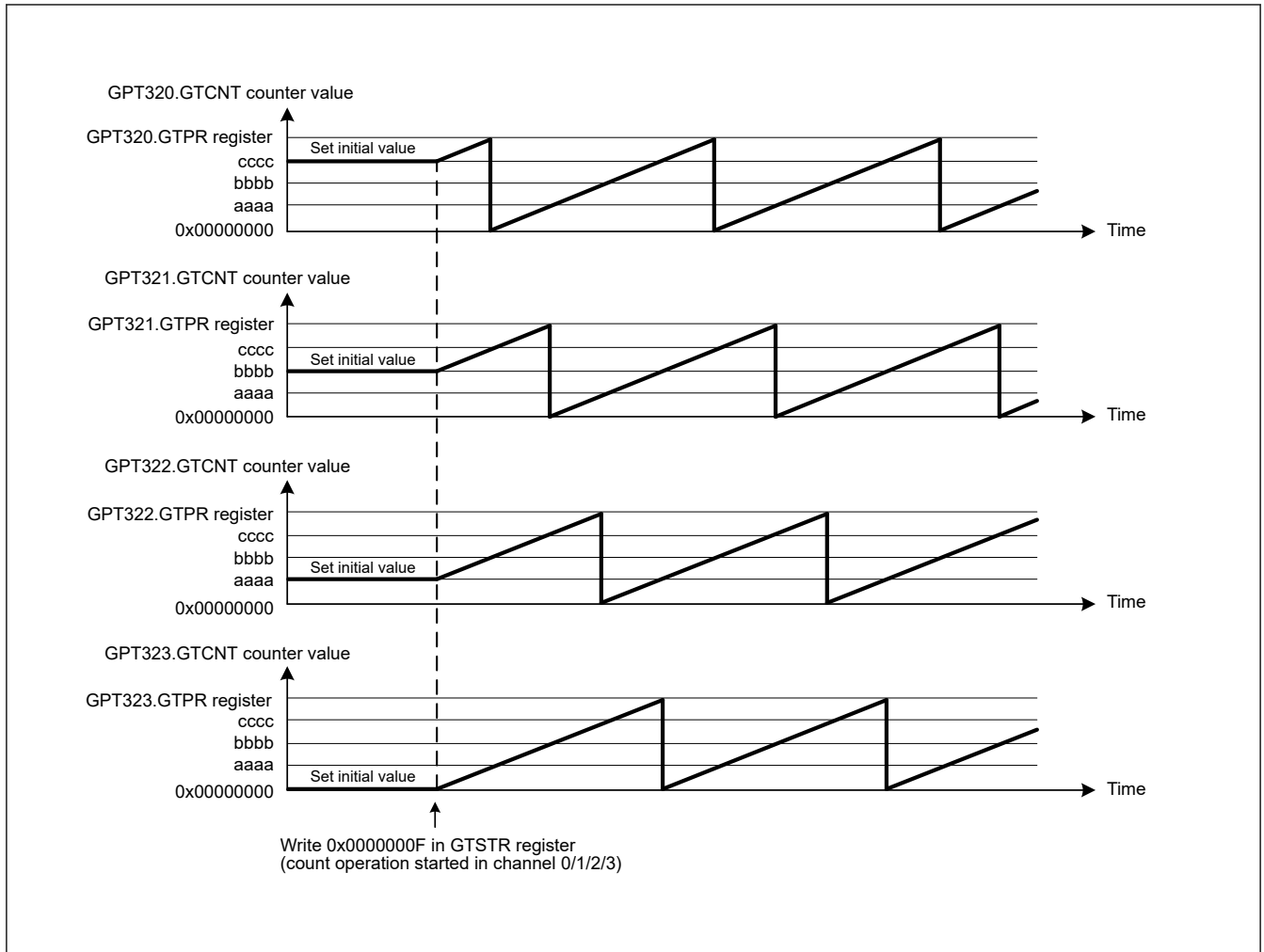


Figure 21.39 Example of software phase start with the same count cycle (GTPR register value)

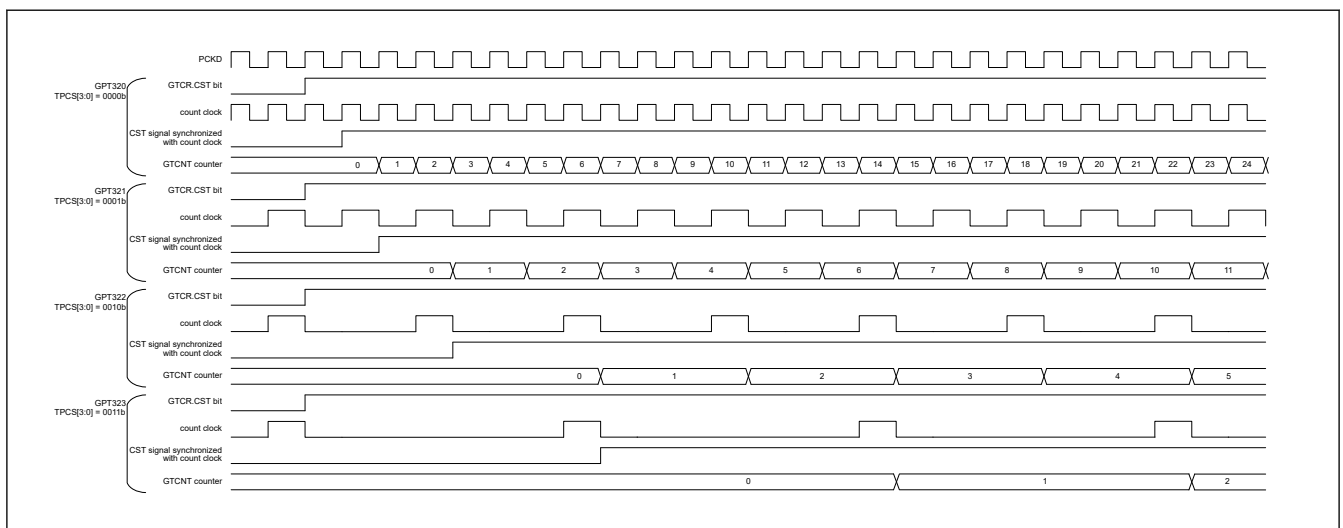


Figure 21.40 Example of Simultaneous Start Operation by Software (with Different Count Period)

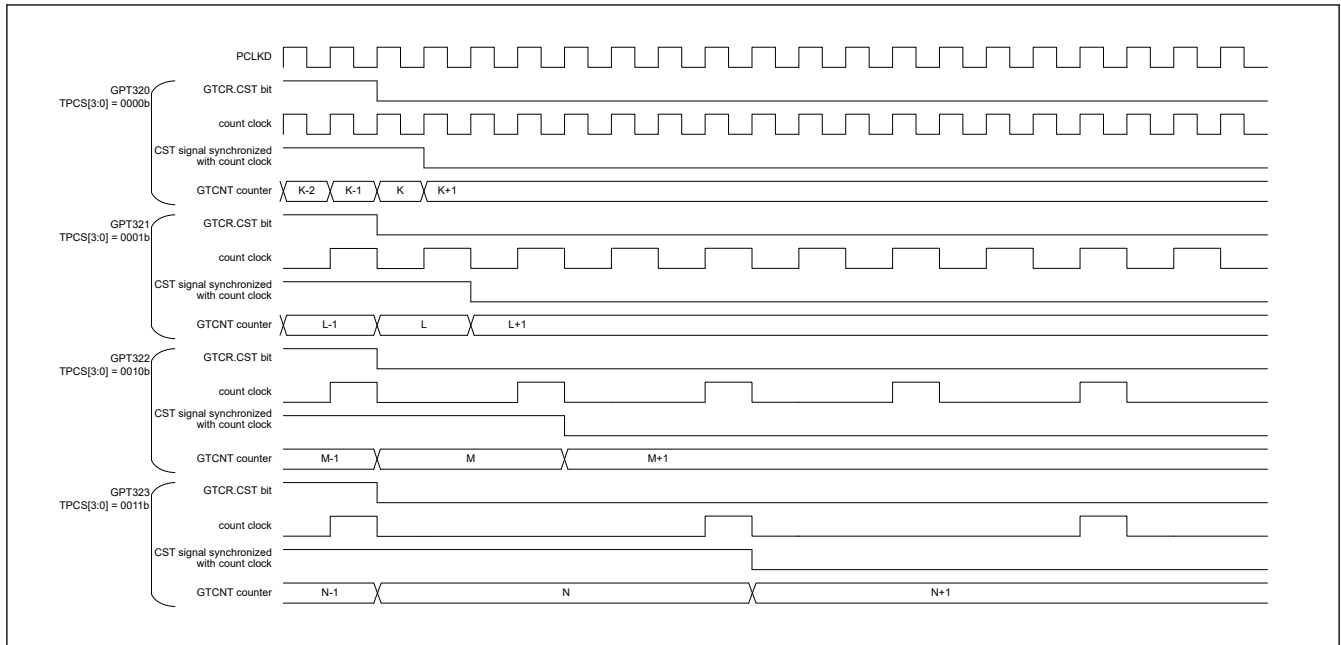


Figure 21.41 Example of Simultaneous Stop Operation by Software (with Different Count Period)

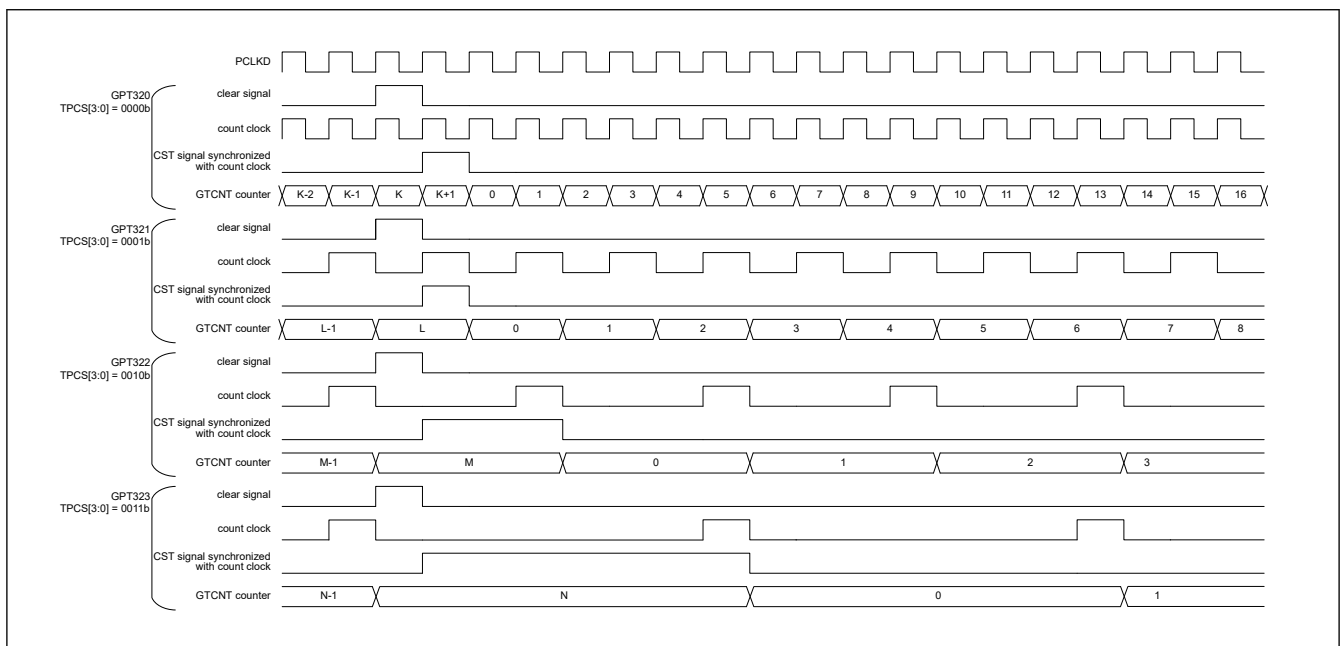


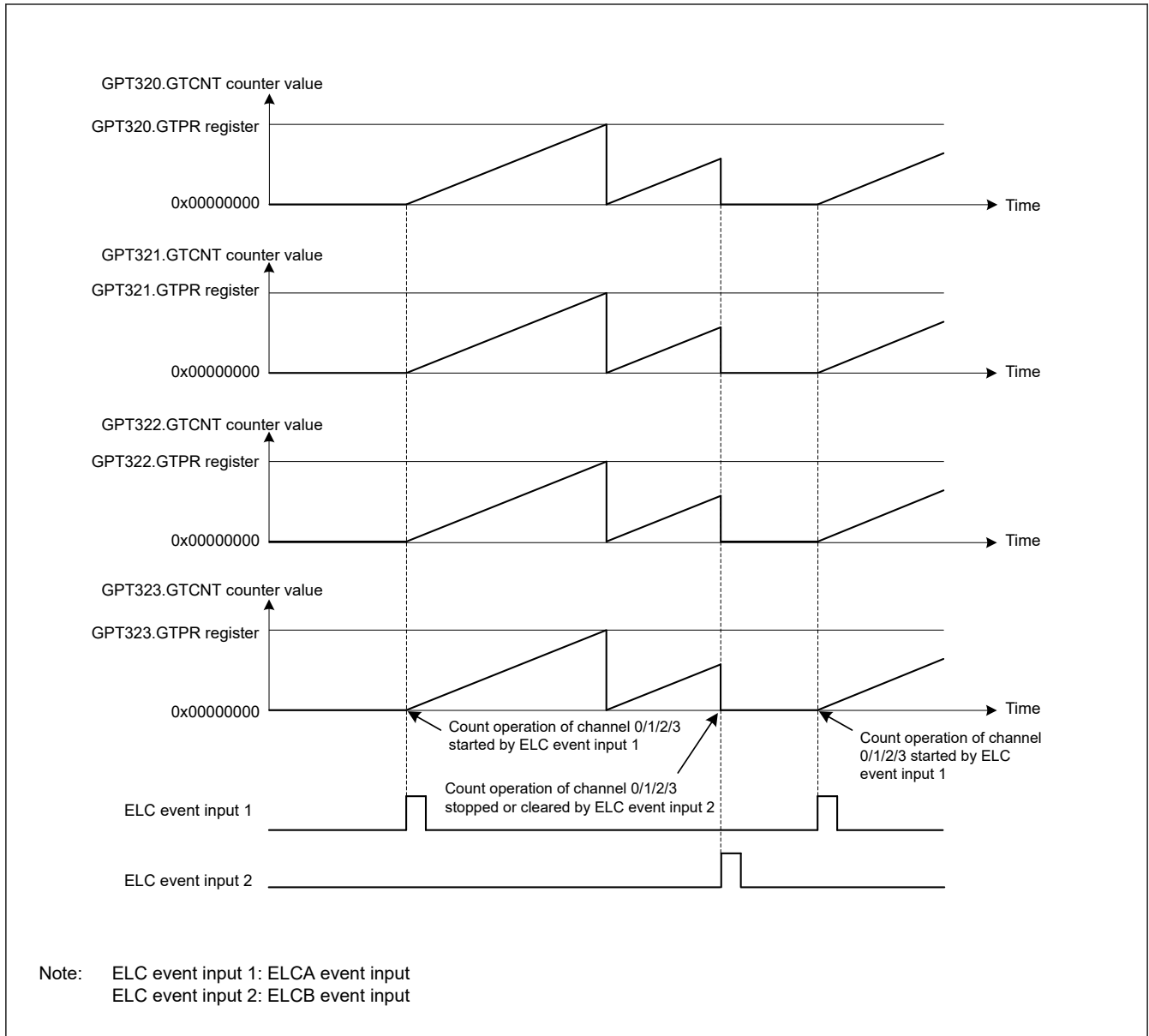
Figure 21.42 Example of Simultaneous Clearing Operation by Software (with Different Count Period)

### 21.3.8.2 Synchronized Operation by Hardware

The counters for multiple channels can be started, stopped, and cleared simultaneously by ELC event input.

Figure 21.43 shows an example of a simultaneous start, stop, and clear operation by a hardware source. Table 21.29 shows the setting example.





**Figure 21.43 Example of a simultaneous start, stop, and clear by a hardware source with the same count cycle (GTPR register value)**

**Table 21.29 Example setting for simultaneous start by a hardware source (1 of 2)**

No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In Figure 21.43, 000b (saw-wave PWM mode) is set.
2	Set count direction	Select the count direction (up or down) with the GTUDDTYC register. In Figure 21.43, after 11b is set in the GTUDDTYC[1:0] bits, 01b is set in the GTUDDTYC[1:0] bits (up-counting).
3	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
4	Set cycle	Set the cycle in the GTPR register.
5	Set initial value for counter	Set the initial value in the GTCNT counter. In Figure 21.43, 0x00000000 is set.
6	Set hardware count start	Select a hardware source for starting count operation with the GTSSR register, and wait for count start by the hardware source. In Figure 21.43, GTSSR.SSELCA = 1.

**Table 21.29 Example setting for simultaneous start by a hardware source (2 of 2)**

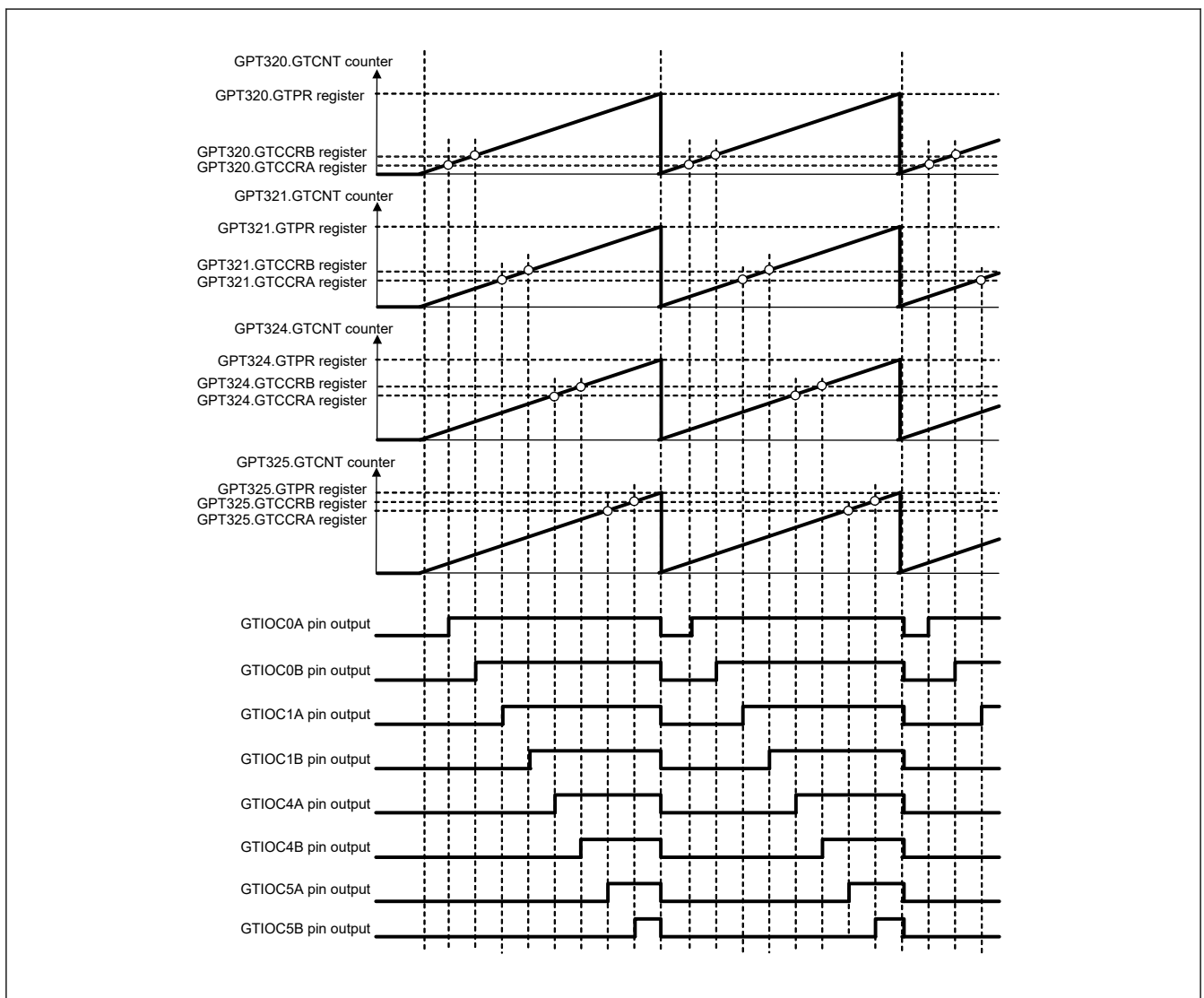
No.	Step Name	Description
7	Set hardware count stop	Select a hardware source for stopping count operation with the GTPSR register, and wait for count stop by the hardware source. In <a href="#">Figure 21.43</a> , GTPSR.PSELCB = 1.
8	Set hardware count clear	Select a hardware source for clearing count operation with the GTCSR register, and wait for count clear by the hardware source. In <a href="#">Figure 21.43</a> , GTCSR.CSELCB = 1.
9	Set hardware source operation	Set operation of the hardware source selected in the GTSSR, GTPSR, or GTCSR registers, and start, stop, or clear counting. In <a href="#">Figure 21.43</a> , ELCA input and ELCB input are set.

### 21.3.9 PWM Output Operation Examples

#### (1) Synchronized PWM output

The GPT outputs  $10 \times 2$  phases of linked PWM waveforms for a maximum of  $GPT \times 10$  channels.

[Figure 21.44](#) shows an example in which four channels perform synchronized operation in saw-wave PWM mode and eight phases of PWM waveforms are output. The GTIOCnA is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCnB is set so that it outputs low as the initial value, high at a GTCCRB compare match, and low at the cycle end.



**Figure 21.44 Example of synchronized PWM output**

(2) 3-phase saw-wave complementary PWM output

Figure 21.45 shows an example in which three channels perform synchronized operation in saw-wave PWM mode and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, high at a GTCCRA compare match, and low at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, low at a GTCCRB compare match, and high at the cycle end.

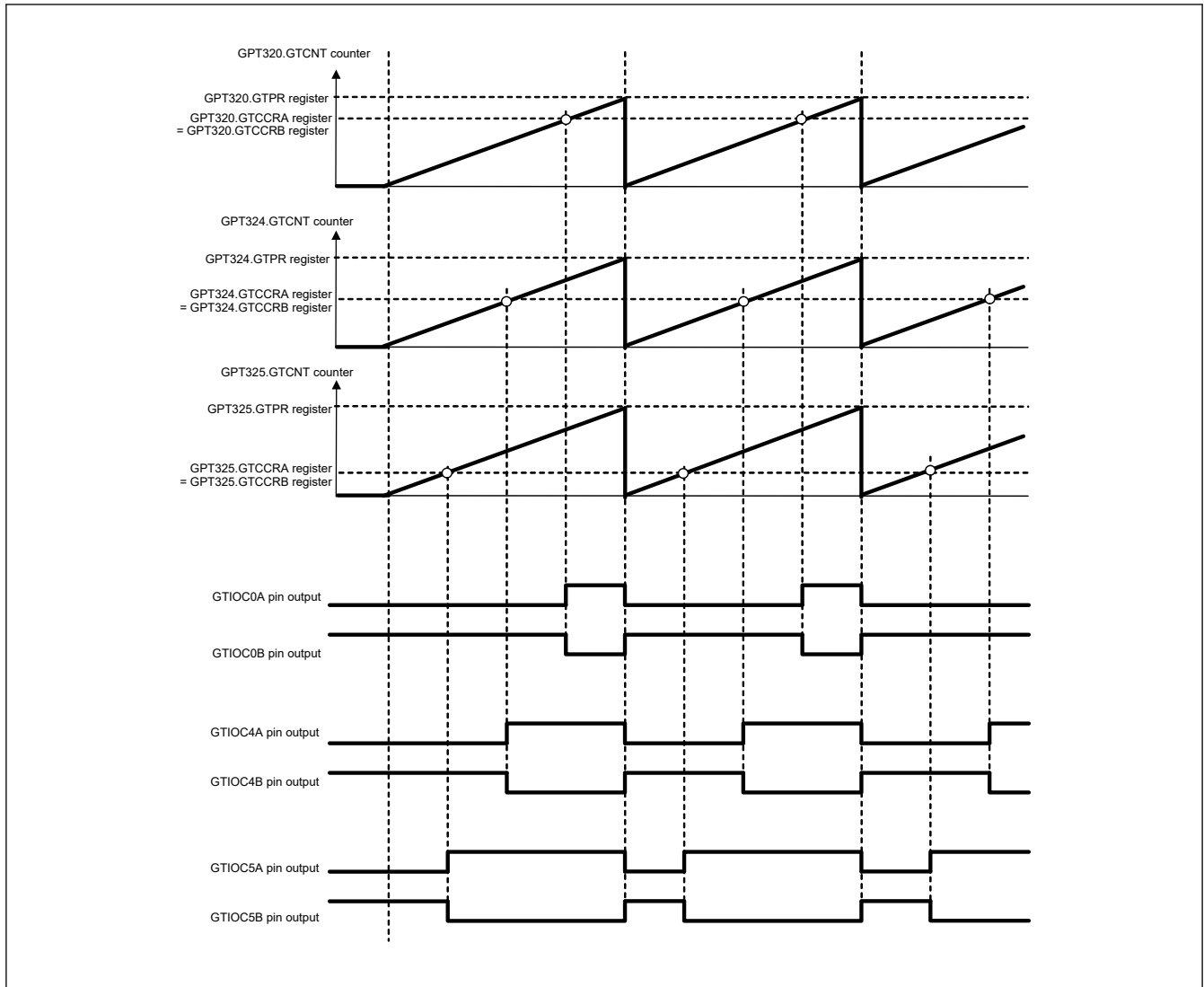
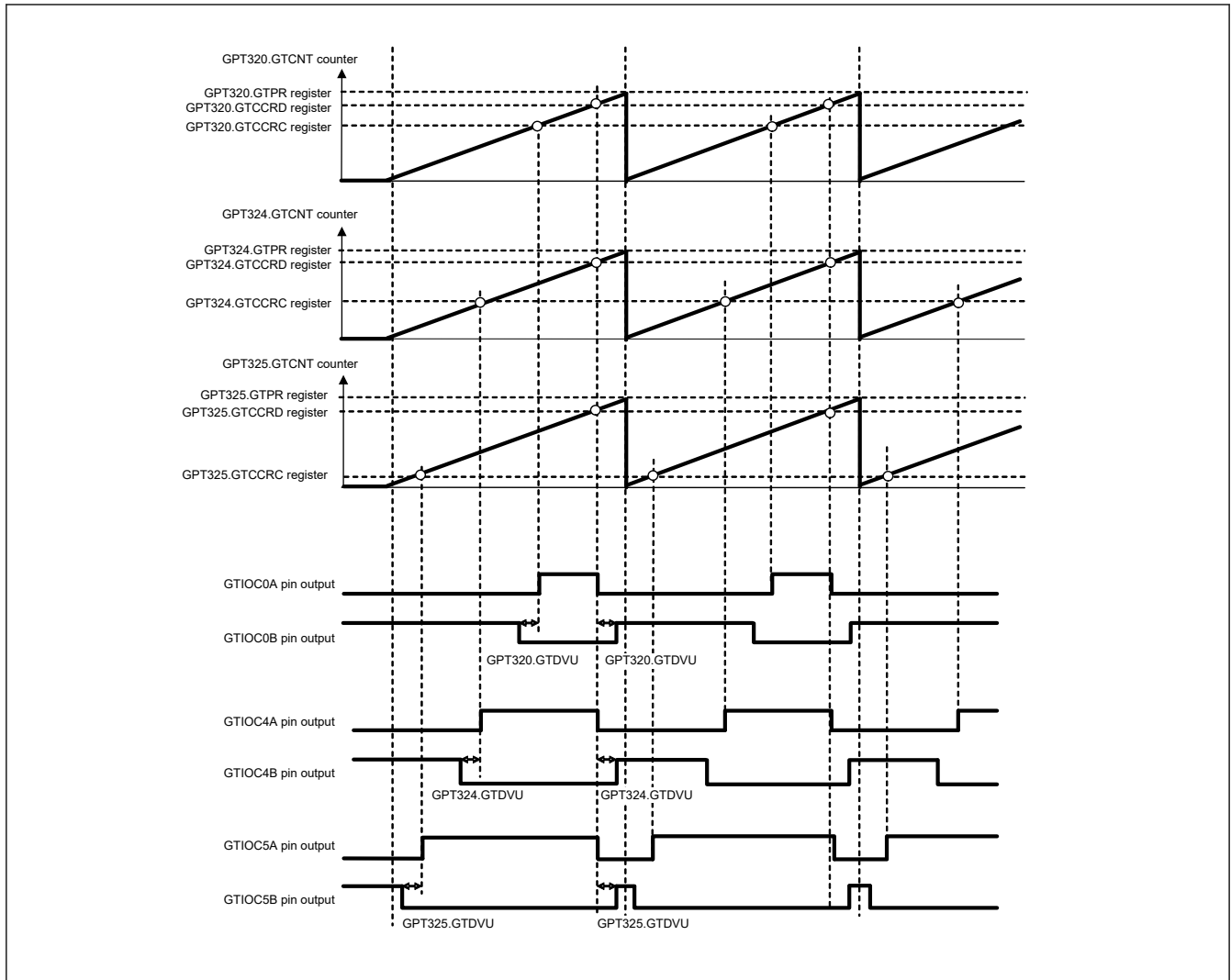


Figure 21.45 Example of 3-phase saw-wave complementary PWM output

(3) 3-phase saw-wave complementary PWM output with automatic dead time setting

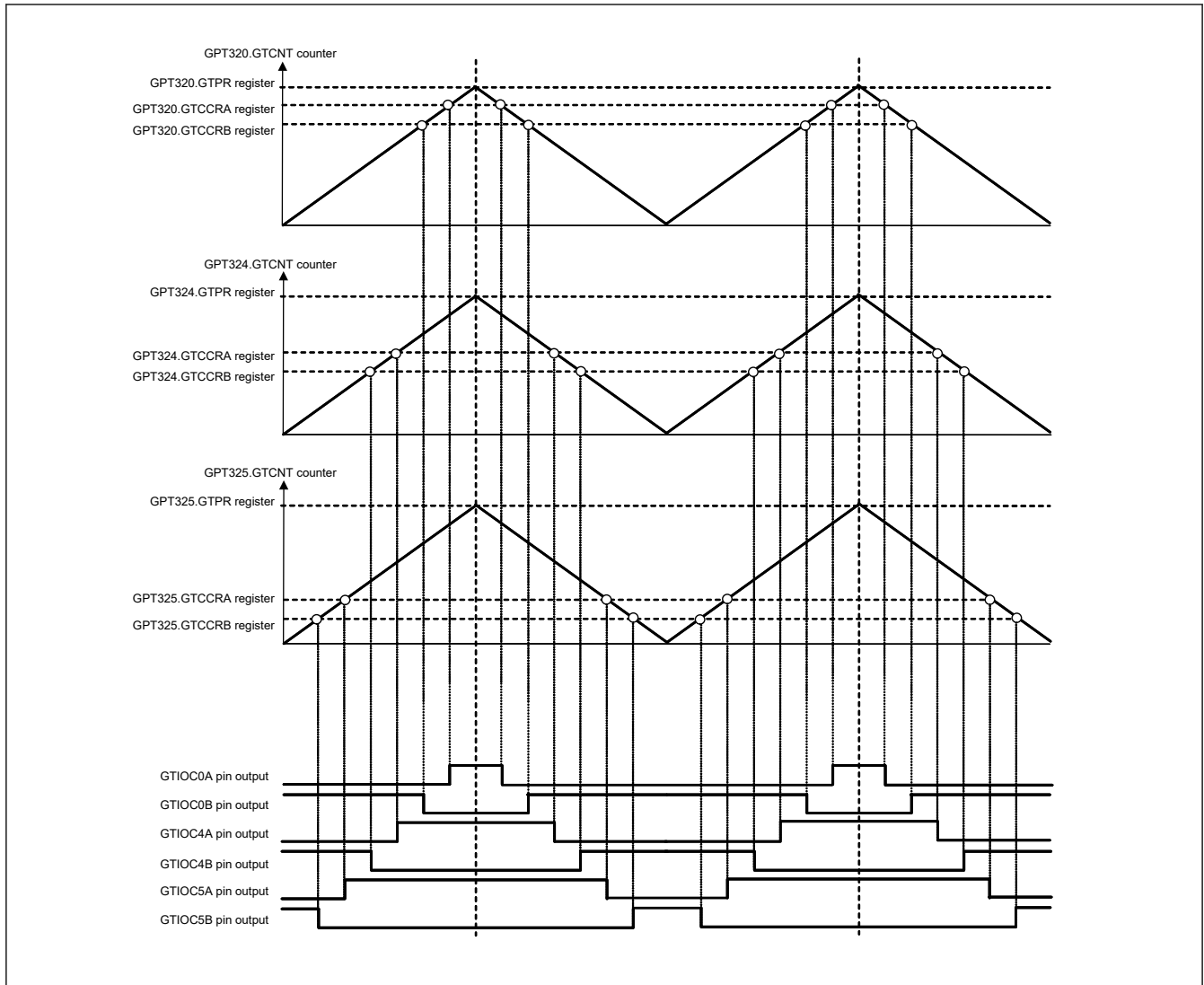
Figure 21.46 shows an example in which three channels perform synchronized operation in saw-wave one-shot pulse mode with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTIOCnB compare match, and retains the output at the cycle end.



**Figure 21.46 Example of 3-phase saw-wave complementary PWM output with automatic dead time setting**

**(4) 3-phase triangle-wave complementary PWM output**

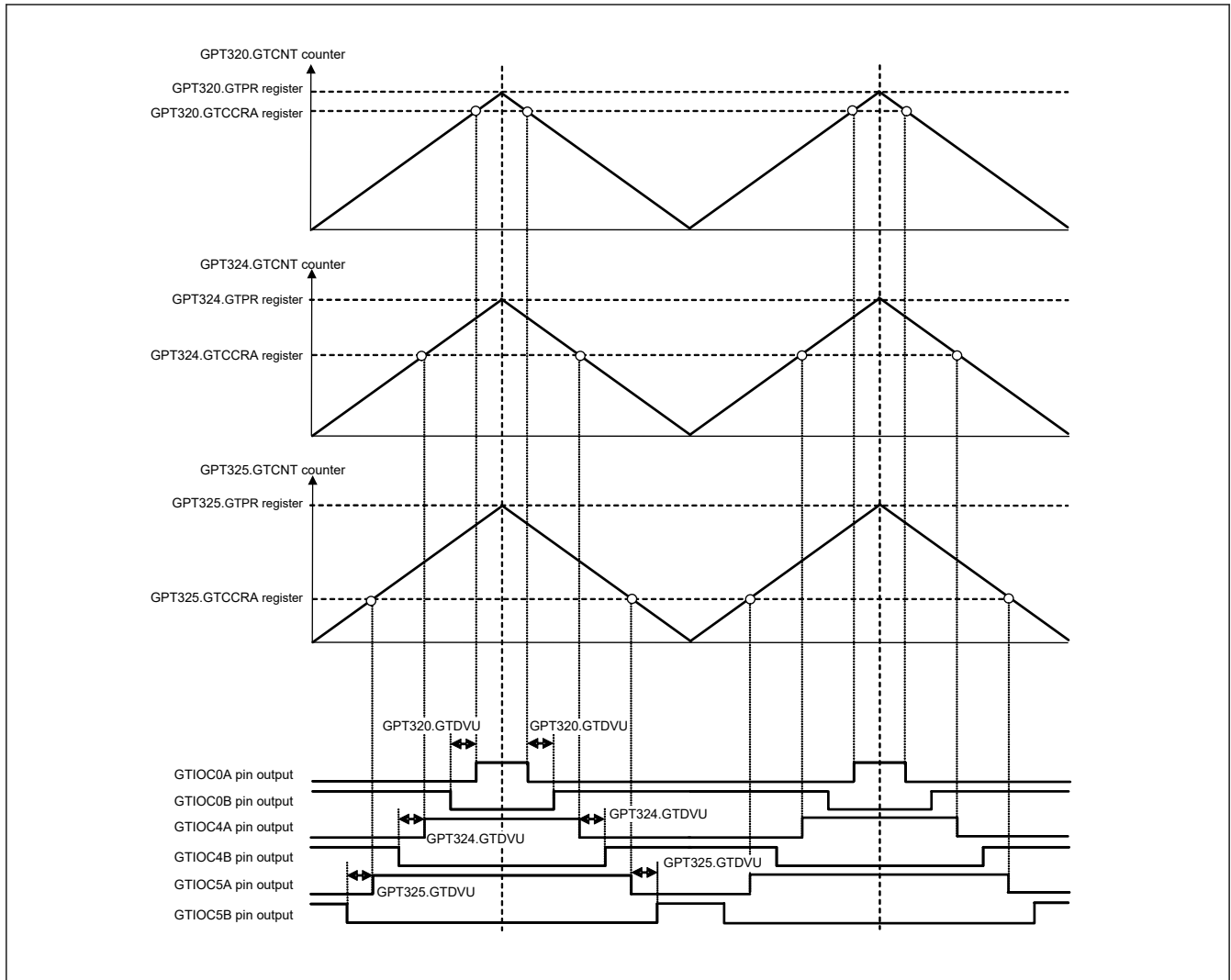
Figure 21.47 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.



**Figure 21.47 Example of 3-phase triangle-wave complementary PWM output**

(5) 3-phase triangle-wave complementary PWM output with automatic dead time setting

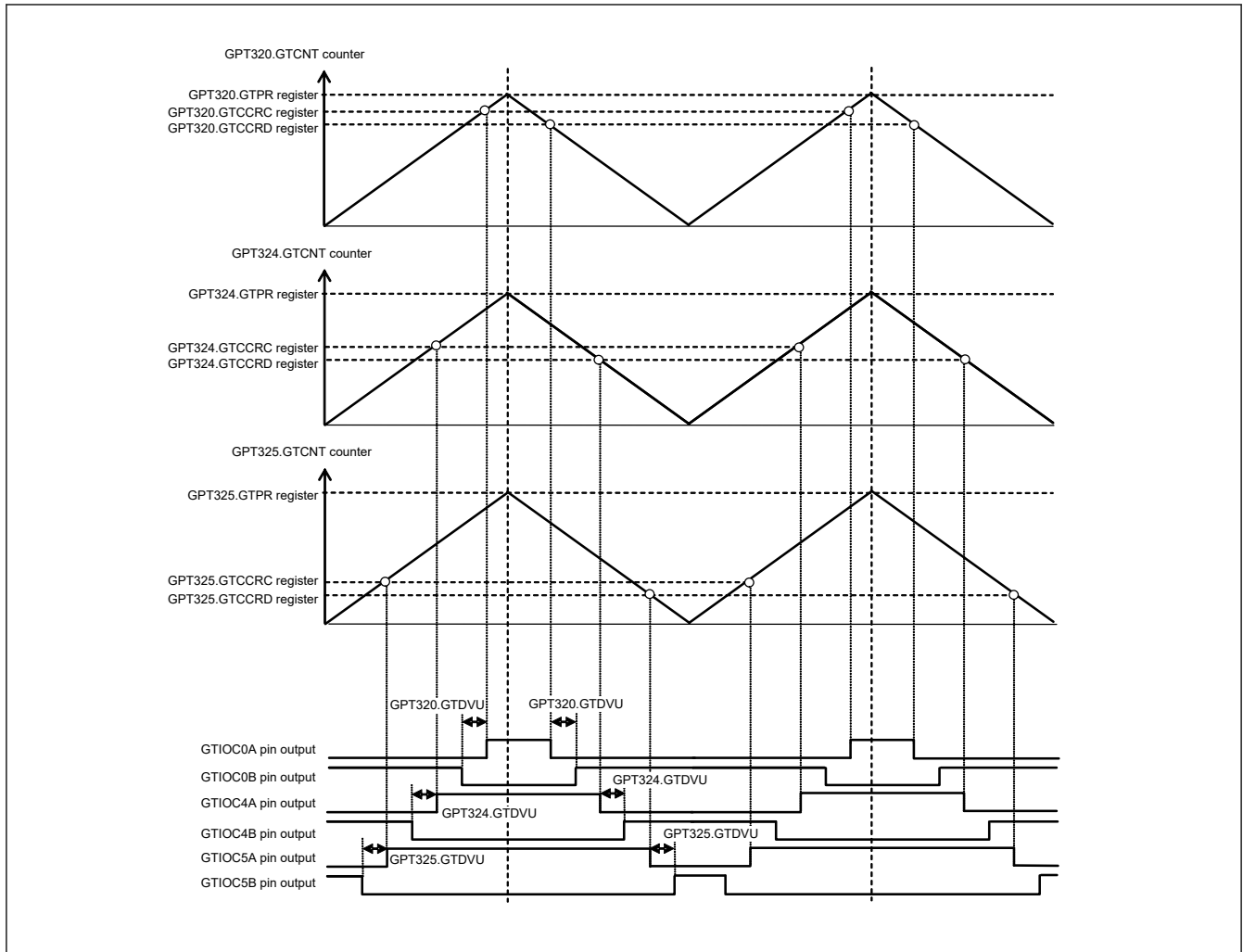
Figure 21.48 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 1 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB pin is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.



**Figure 21.48 Example of 3-phase triangle-wave complementary PWM output with automatic dead time setting**

(6) 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting

Figure 21.49 shows an example in which three channels perform synchronized operation in triangle-wave PWM mode 3 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA is set so that it outputs low as the initial value, toggles the output at a GTCCRA compare match, and retains the output at the cycle end. The GTIOCnB is set so that it outputs high as the initial value, toggles the output at a GTCCRB compare match, and retains the output at the cycle end.



**Figure 21.49 Example of 3-phase asymmetric triangle-wave complementary PWM output with automatic dead time setting**

### 21.3.10 Period Count Function

By setting the GTPC register, the end of period can be counted.

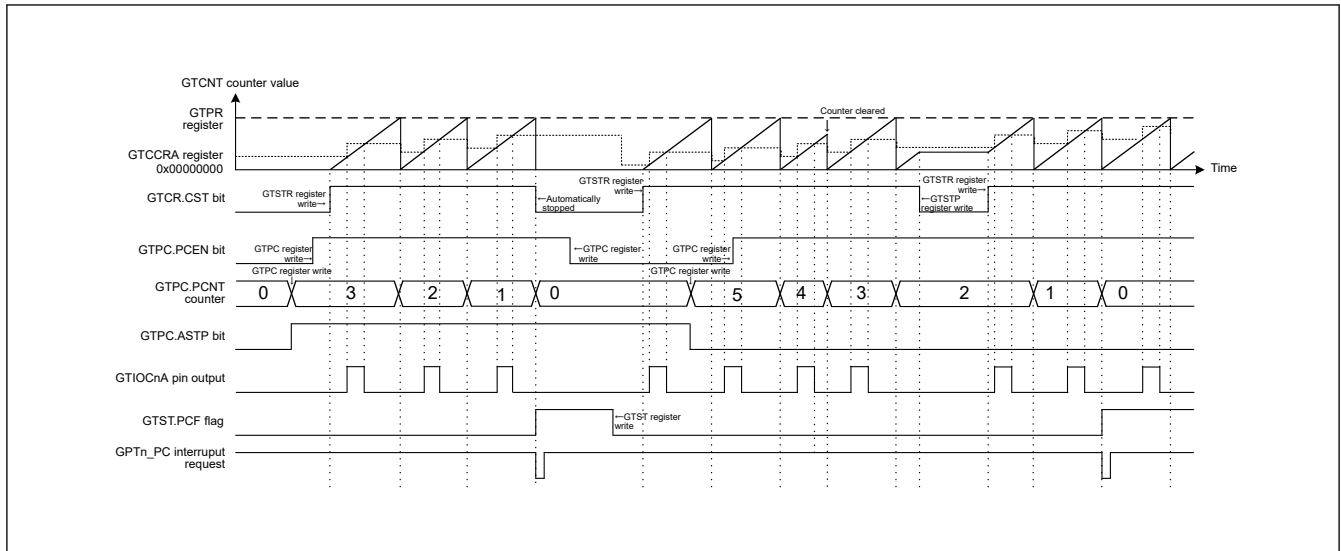
The number of period to be counted should be set into the GTPC.PCNT counter when the GTPC.PCEN bit is 0. When the PCEN bit is 1, the PCNT counter can be read, but writing is disabled. When the PCEN bit is 1, down-counting is performed at the end of period. When the PCNT counter is 1 at the end of period, it becomes 0 and counting is stopped to finish the period count function. At that time, the GTST.PCF flag is set, and the period count function finish interrupt request GPTn\_PC is generated. When the GTPC.ASTP bit is 1, the GTCNT counter is also stopped at the same time that the period count function is finished.

When the GTCNT counter is stopped while period count function is enabled, the PCNT counter keeps its value. When the GTCNT counter restarts counting and the PCEN bit is 1, the PCNT counter restarts down-counting from the hold value.

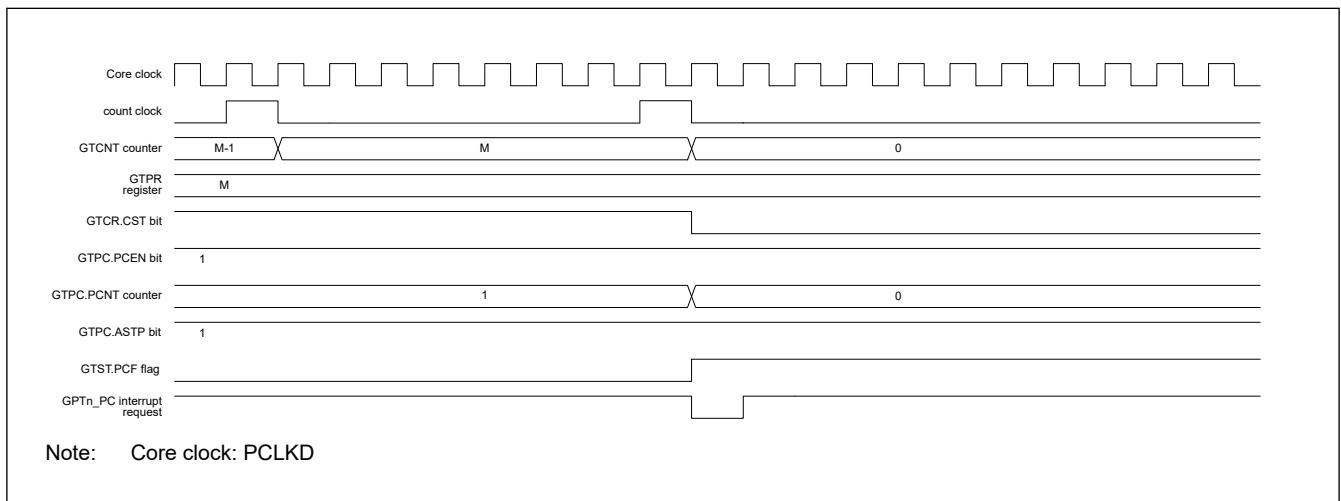
When the PCEN bit is changed from 0 to 1 while the PCNT counter is 0 and the ASTP bit is 1, the GTCNT counter is stopped at the count clock immediately after that.

When either GTSECR.SPCE bit or GTSECR.SPCD bit is set to 1, the PCEN bit in the channels set to 1 by the GTSECSR register is simultaneously set the value to enable or disable the period count function for multiple channels.

Figure 21.50 and Figure 21.51 show examples of PWM cycle count function.



**Figure 21.50 Example of PWM cycle count function (saw-wave one-shot pulse mode)**



**Figure 21.51 Example of the timing of operations for PWM cycle count function (saw-wave one-shot pulse mode, up-counting)**

### 21.3.11 Phase Counting Function

The phase difference between the GTIOcNA and GTIOcNB pin (n = 0 to 5, 10 to 13) inputs is detected and the associated GTCNT counts up or counts down. The detectable phase difference is available in any combination with the relationship between the edge and the level of GTIOcNA and GTIOcNB pin inputs being set in the GTUPSR and GTDNSR registers. For details on count operation, see [section 21.3.1.1. Counter operation](#).

[Figure 21.52](#) to [Figure 21.61](#) show an example of phase counting modes 1 to 5 operation when the GTIOcNA, GTIOcNB pins are used. [Table 21.30](#) to [Table 21.39](#) show conditions of up-counting or down-counting and list settings for the GTUPSR and GTDNSR registers which is corresponding to [Figure 21.52](#) to [Figure 21.61](#).



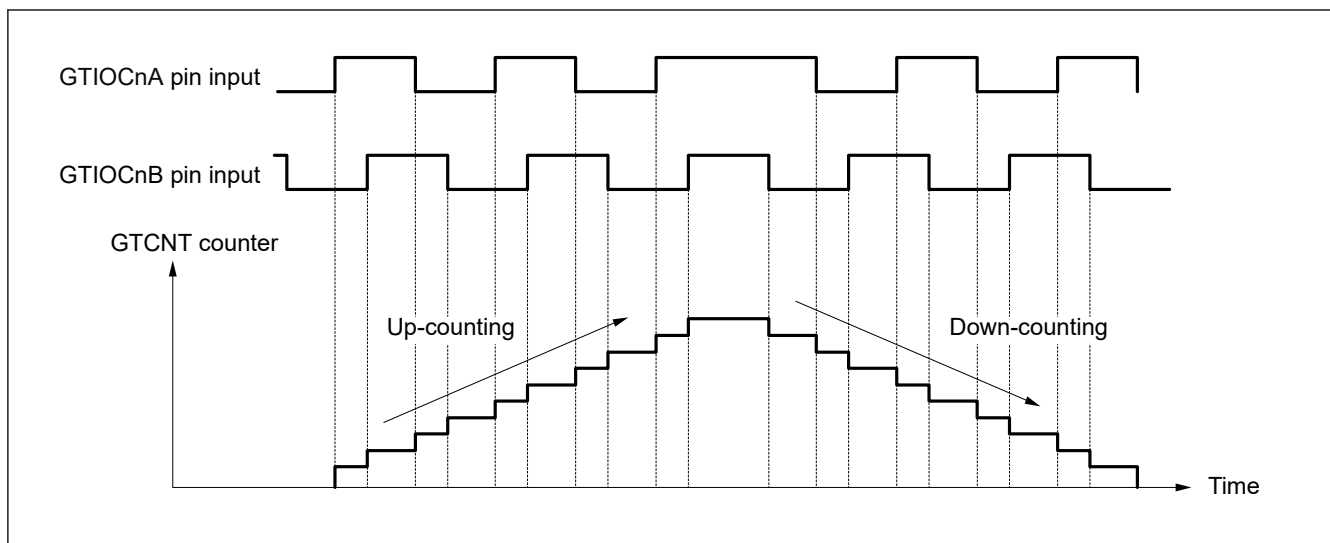












Figure 21.52 Example of phase counting mode 1

Table 21.30 Conditions of up-counting/down-counting in phase counting mode 1

 : Rising edge  
 : Falling edge

GTIOcNA pin input	GTIOcNB pin input	Operation	Register setting
High		Up-counting	GTUPSR = 0x00006900 GTDNSR = 0x00009600
Low			
	Low		
	High		
High		Down-counting	
Low			
	High		
	Low		

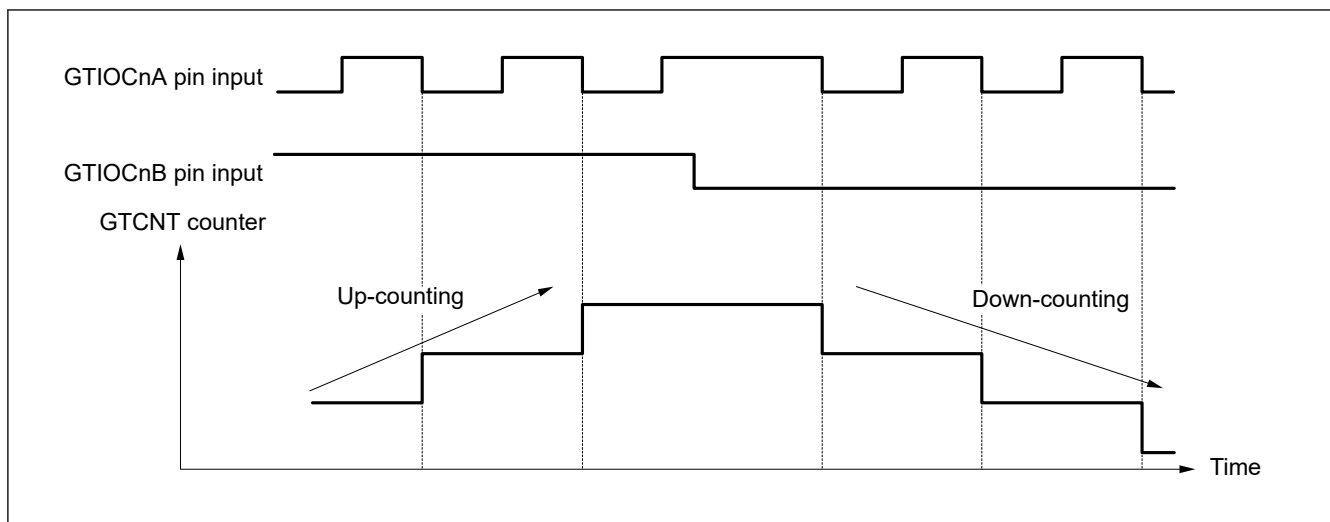












Figure 21.53 Example of phase counting mode 2 (A)

Table 21.31 Conditions of up-counting/down-counting in phase counting mode 2 (A)

 : Rising edge  
 : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000800 GTDNSR = 0x00000400
Low			
	Low		
	High	Up-counting	
High		Not counting	
Low			
	High		
	Low	Down-counting	

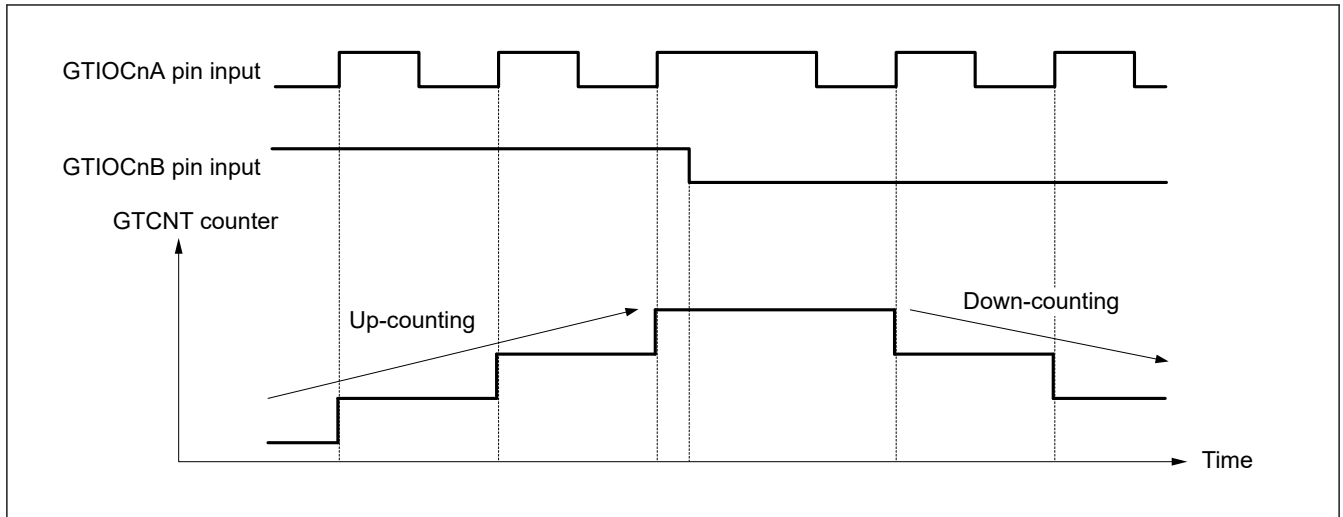












Figure 21.54 Example of phase counting mode 2 (B)

Table 21.32 Conditions of up-counting/down-counting in phase counting mode 2 (B)

 : Rising edge  
 : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000200 GTDNSR = 0x00000100
Low			
	Low	Down-counting	
	High	Not counting	
High			
Low		Up-counting	
	High		
	Low	Not counting	

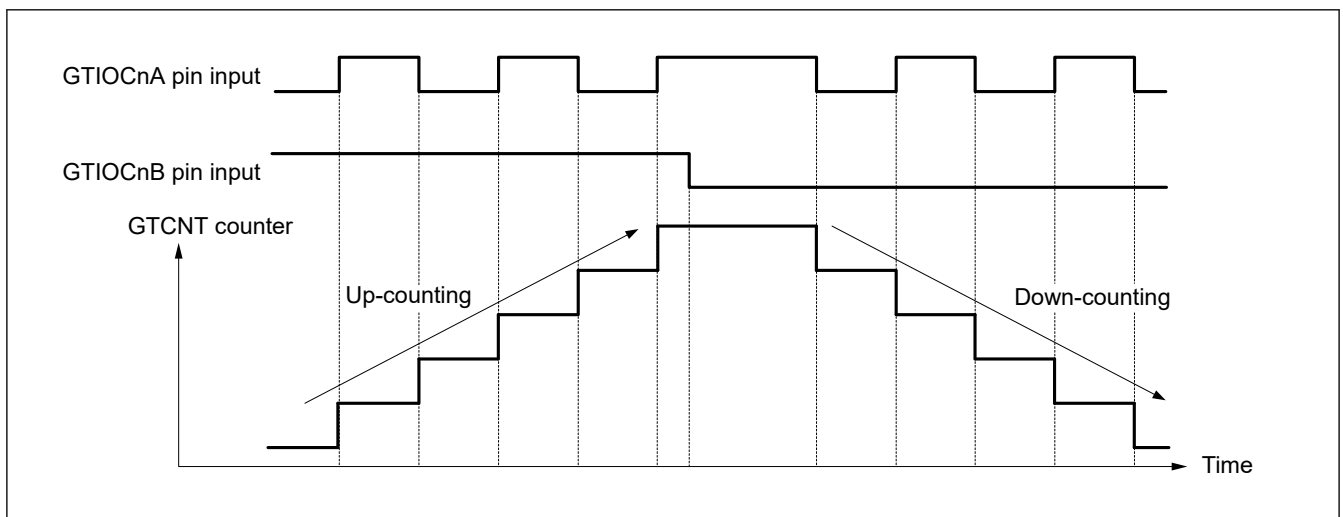










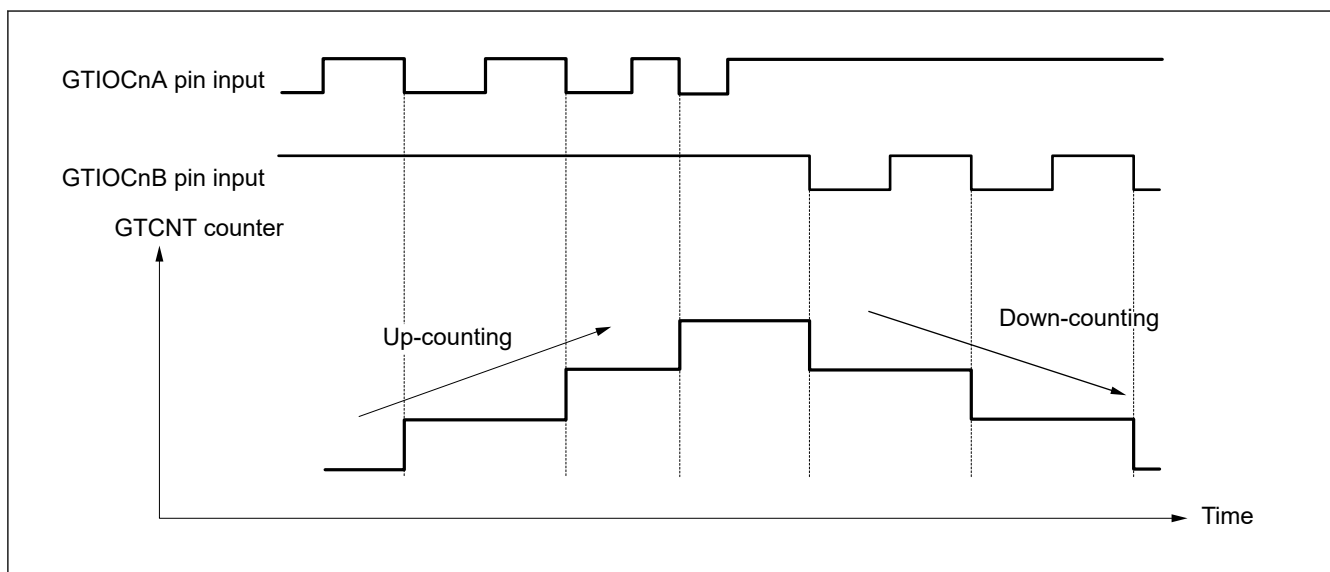


Figure 21.55 Example of phase counting mode 2 (C)

**Table 21.33 Conditions of up-counting/down-counting in phase counting mode 2 (C)**



 : Rising edge  
 : Falling edge









GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000A00 GTDNSR = 0x00000500
Low			
	Low	Down-counting	
	High	Up-counting	
High		Not counting	
Low			
	High	Up-counting	
	Low	Down-counting	

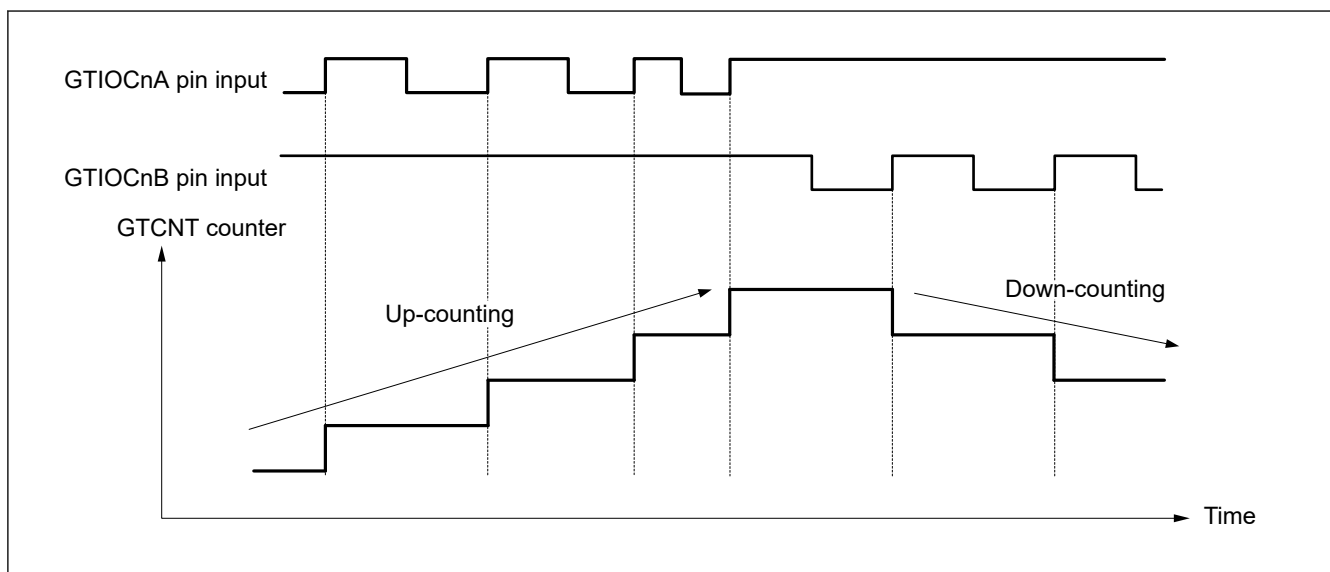


**Figure 21.56 Example of phase counting mode 3 (A)**

**Table 21.34 Conditions of up-counting/down-counting in phase counting mode 3 (A)**



 : Rising edge  
 : Falling edge









GTIOcNA pin input	GTIOcNB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000800 GTDNSR = 0x00000800
Low			
	Low		
	High	Up-counting	
High		Down-counting	
Low		Not counting	
	High		
	Low		

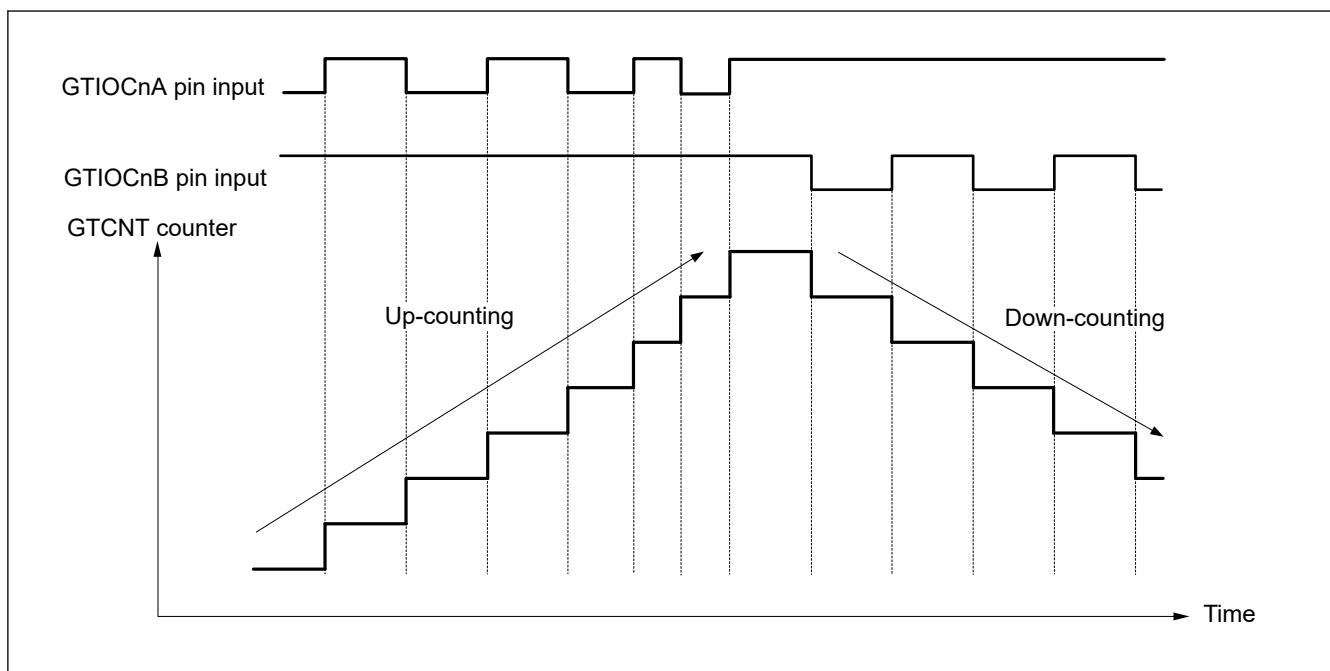


**Figure 21.57 Example of phase counting mode 3 (B)**

**Table 21.35 Conditions of up-counting/down-counting in phase counting mode 3 (B)**



 : Rising edge  
 : Falling edge









GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Down-counting	GTUPSR = 0x00000200 GTDNSR = 0x00002000
Low		Not counting	
	Low		
	High		
High			
Low			
	High	Up-counting	
	Low	Not counting	

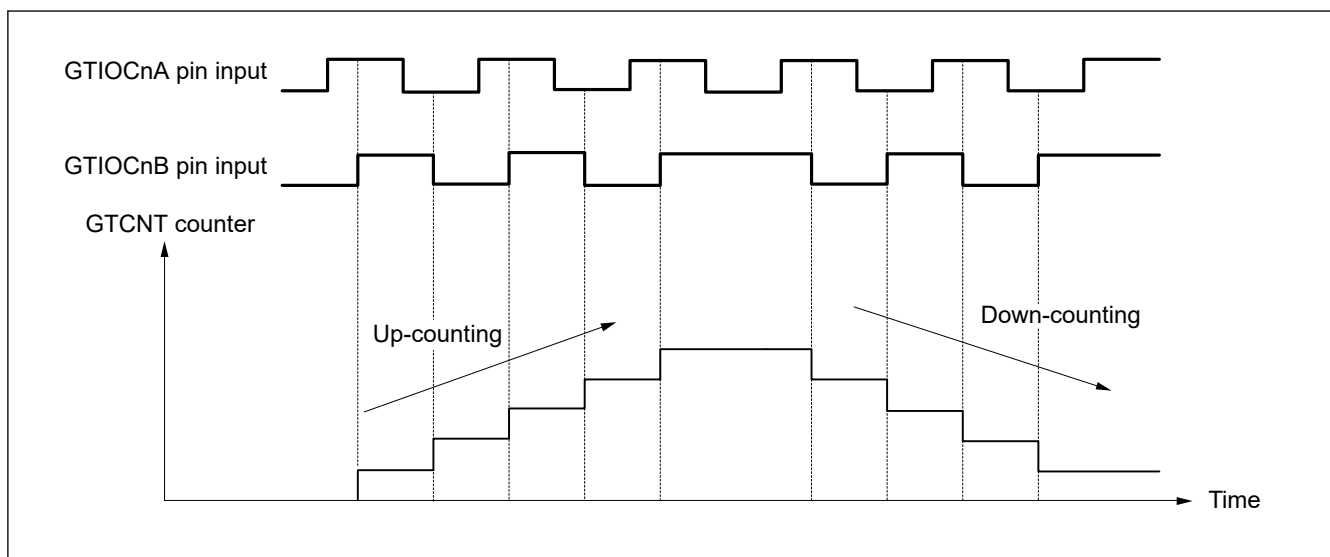


**Figure 21.58 Example of phase counting mode 3 (C)**

**Table 21.36 Conditions of up-counting/down-counting in phase counting mode 3 (C)**



 : Rising edge  
 : Falling edge









GTIOcNA pin input	GTIOcNB pin input	Operation	Register setting
High		Down-counting	GTUPSR = 0x00000A00 GTDNSR = 0x0000A000
Low		Not counting	
	Low		
	High	Up-counting	
High		Down-counting	
Low		Not counting	
	High	Up-counting	
	Low	Not counting	

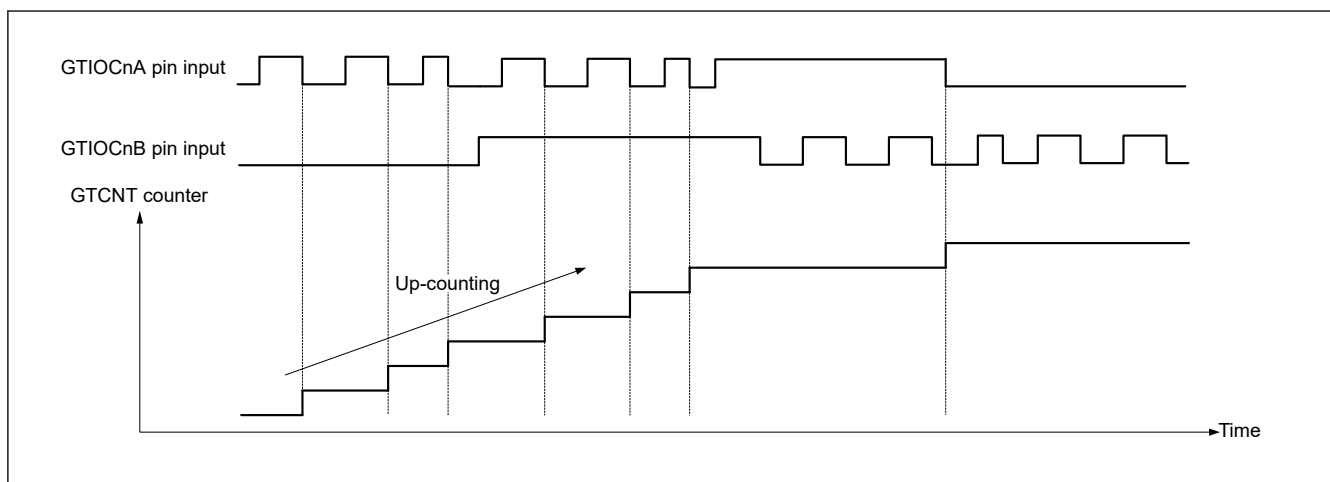


**Figure 21.59 Example of phase counting mode 4**

**Table 21.37 Conditions of up-counting/down-counting in phase counting mode 4**

 : Rising edge  
 : Falling edge



GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Up-counting	GTUPSR = 0x00006000 GTDNSR = 0x00009000
Low			
	Low	Not counting	
	High		
High		Down-counting	
Low			
	High	Not counting	
	Low		











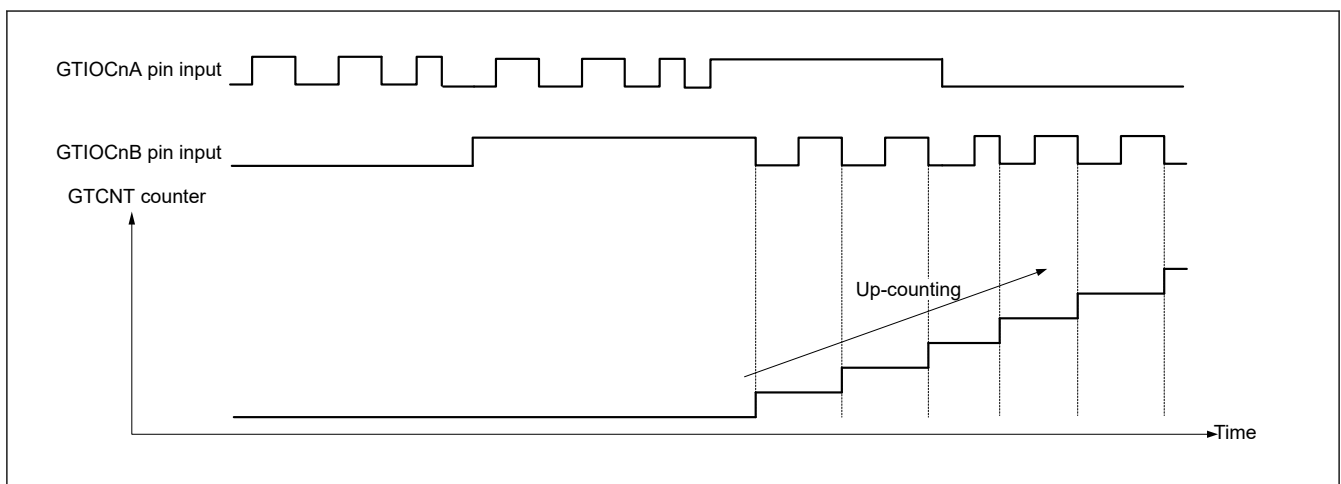
**Figure 21.60 Example of phase counting mode 5 (A)**



**Table 21.38 Conditions of up-counting/down-counting in phase counting mode 5 (A)**



 : Rising edge  
 : Falling edge









GTIOcNA pin input	GTIOcNB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x00000C00 GTDNSR = 0x00000000
Low			
	Low		
	High	Up-counting	
High		Not counting	
Low			
	High		
	Low	Up-counting	



**Figure 21.61 Example of phase counting mode 5 (B)**

**Table 21.39** Conditions of up-counting/down-counting in phase counting mode 5 (B)

 : Rising edge  
 : Falling edge

GTIOCnA pin input	GTIOCnB pin input	Operation	Register setting
High		Not counting	GTUPSR = 0x0000C000 GTDNSR = 0x00000000
Low		Up-counting	
	Low	Not counting	
	High		
High		Up-counting	
Low		Not counting	
	High		
	Low		

### 21.3.12 Inter Channel Logical Operation Function

The logical operation function between compare match outputs can be performed.

Figure 21.62 shows the block diagram of inter channel logical operation.

To prevent hazard to the GPT output, the signal after logical operation is latched with PCLKD. After latching, the output disable control is performed.

When the logical operation function which causes the delay of 1 PCLKD is selected, the output enable signal is also delayed with 1 PCLKD and input to the output disable control.

When the same signal (C = A or D = B) to operate logical function AND, OR, EXOR and NOR is selected, C or D is treated as 1. In the case of GTIOCnA pin output, when A of same channel is selected for C, the result of AND is A, the result of OR is 1, the result of EXOR is NOT A, and the result of NOR is 0.

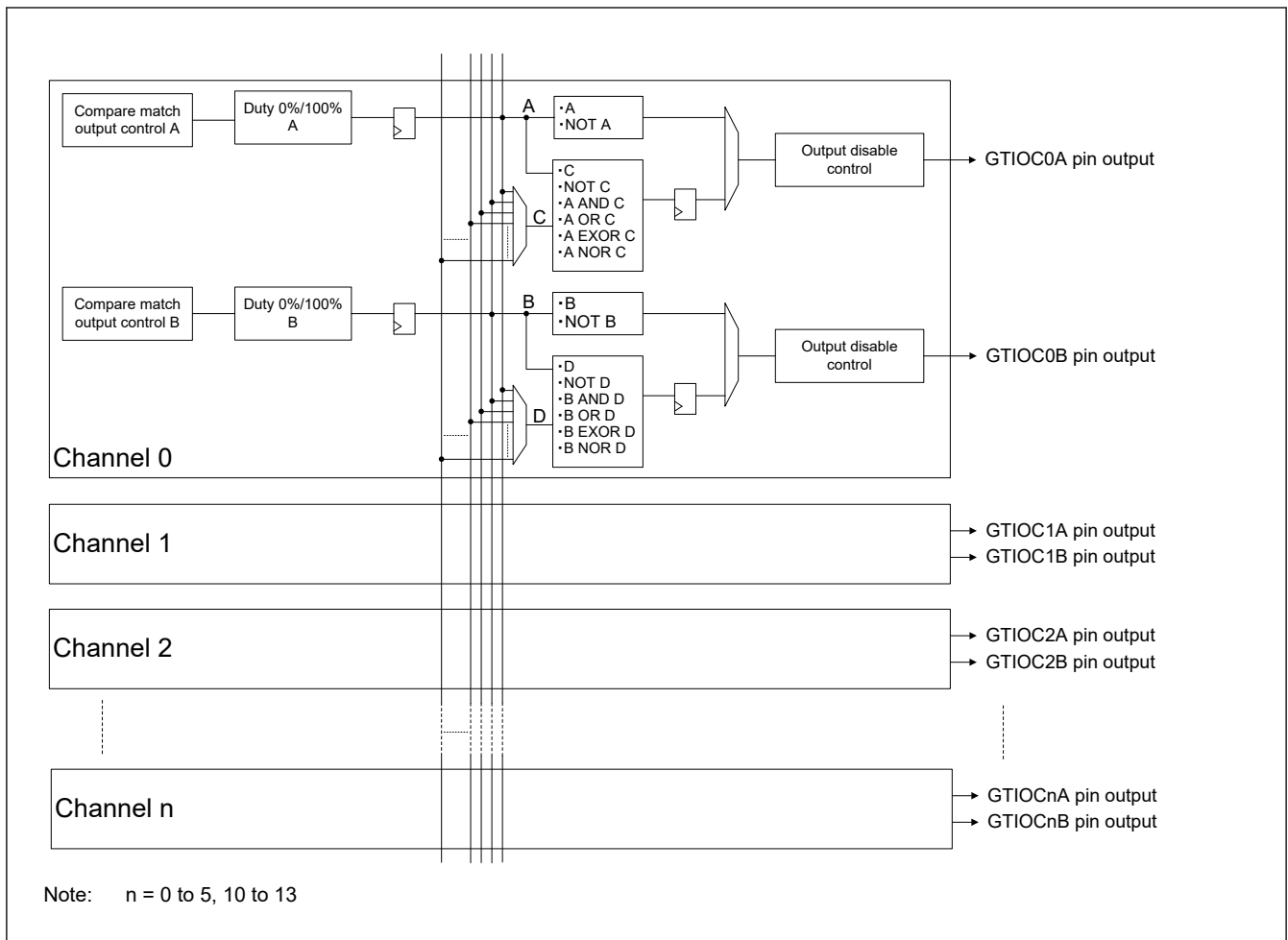


Figure 21.62 Block Diagram of Inter Channel Logical Operation

Figure 21.63 shows an example of inter channel logical operation.

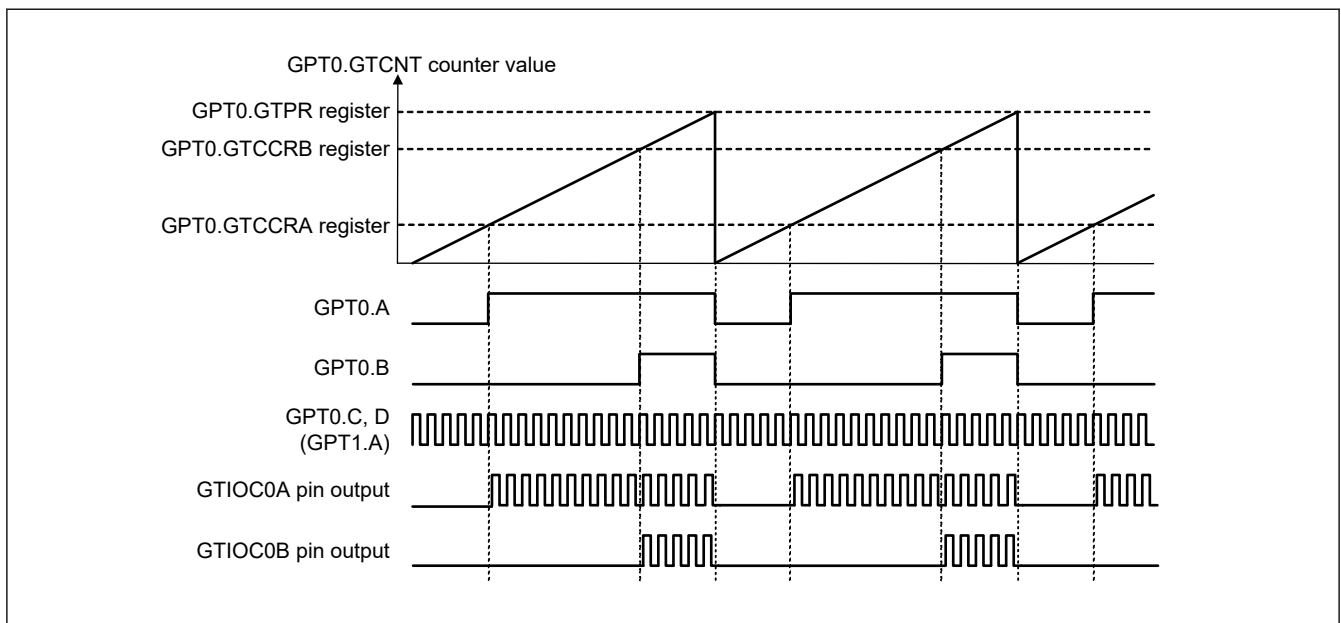


Figure 21.63 Example of Inter Channel Logical Operation

## 21.4 Interrupt Sources

### 21.4.1 Interrupt Sources

The GPT provides the following interrupt sources:

- GTCCR input capture/compare match
- GTCNT counter overflow (GTPR compare match)/underflow.
- Period count function finish

Each interrupt source has its own status flag. When an interrupt source signal is generated, the associated status flag in GTST is set to 1, and an interrupt request is generated. The associated status flag in GTST can be cleared by writing 0. If flag set and flag clear occur at the same time, flag clear takes priority over flag set. These flags are automatically updated by the internal state. The Interrupt Controller Unit can change the relative channel priorities. However, the priority within a channel is fixed. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

[Table 21.40](#) lists the GPT interrupt sources.

**Table 21.40** Interrupt sources

Channel	Name	Interrupt source	Interrupt flag	DTC activation
n = 0 to 5	GPTn_CCMPA	GPT32n.GTCCRA input capture/compare match	GTST[0] (TCFA)	Possible
	GPTn_CCMPB	GPT32n.GTCCRB input capture/compare match	GTST[1] (TCFB)	Possible
	GPTn_CMPC	GPT32n.GTCCRC compare match	GTST[2] (TCFC)	Possible
	GPTn_CMPD	GPT32n.GTCCRD compare match	GTST[3] (TCFD)	Possible
	GPTn_CMPE	GPT32n.GTCCRE compare match	GTST[4] (TCFE)	Possible
	GPTn_CMPF	GPT32n.GTCCRF compare match	GTST[5] (TCFF)	Possible
	GPTn_OVF	GPT32n.GTCNT overflow (GPT32n.GTPR compare match)	GTST[6] (TCFPO)	Possible
	GPTn_UDF	GPT32n.GTCNT underflow	GTST[7] (TCFPU)	Possible
	GPTn_PC	Period count function finish (n = 0 to 3)	GTST[31] (PCF)	Possible
n = 10 to 13	GPTn_CCMPA	GPT16n.GTCCRA input capture/compare match	GTST[0] (TCFA)	Possible
	GPTn_CCMPB	GPT16n.GTCCRB input capture/compare match	GTST[1] (TCFB)	Possible
	GPTn_CMPC	GPT16n.GTCCRC compare match	GTST[2] (TCFC)	Possible
	GPTn_CMPD	GPT16n.GTCCRD compare match	GTST[3] (TCFD)	Possible
	GPTn_CMPE	GPT16n.GTCCRE compare match	GTST[4] (TCFE)	Possible
	GPTn_CMPF	GPT16n.GTCCRF compare match	GTST[5] (TCFF)	Possible
	GPTn_OVF	GPT16n.GTCNT overflow (GPT32n.GTPR compare match)	GTST[6] (TCFPO)	Possible
	GPTn_UDF	GPT16n.GTCNT underflow	GTST[7] (TCFPU)	Possible
	GPTn_PC	Period count function finish (n = 10)	GTST[31] (PCF)	Possible

#### (1) GPTn\_CCMPA interrupt (n = 0 to 5, 10 to 13)

An interrupt request is generated under the following conditions:

- When the GTCCRA register functions as a compare match register, the GTCNT counter value matches with the GTCCRA register
- When the GTCCRA register functions as an input capture register, the input-capture signal causes transfer of the GTCNT counter value to the GTCCRA register.

#### (2) GPTn\_CCMPB interrupt (n = 0 to 5, 10 to 13)

An interrupt request is generated under the following conditions:

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches with the GTCCRB register

- When the GTCCRB register functions as an input capture register, the input-capture signal causes transfer of the GTCNT counter value to the GTCCRB register.

### (3) GPTn\_CMPC interrupt (n = 0 to 5, 10 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRC register functions as a compare match register, the GTCNT counter value matches with the GTCCRC register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRC register).

### (4) GPTn\_CMPD interrupt (n = 0 to 5, 10 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRD register functions as a compare match register, the GTCNT counter value matches with the GTCCRD register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] = 10b, 11b (buffer operation with the GTCCRD register).

### (5) GPTn\_CMPE interrupt (n = 0 to 5, 10 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRE register functions as a compare match register, the GTCNT counter value matches with the GTCCRE register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 01b, 10b, 11b (buffer operation with the GTCCRE register).

### (6) GPTn\_CMPF interrupt (n = 0 to 5, 10 to 13)

An interrupt request is generated under the following condition:

- When the GTCCRF register functions as a compare match register, the GTCNT counter value matches with the GTCCRF register.

A compare match is not performed and therefore, an interrupt is not requested in the following conditions:

- GTCR.MD[2:0] = 001b (saw-wave one-shot pulse mode)
- GTCR.MD[2:0] = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] = 10b, 11b (buffer operation with the GTCCRF register).

### (7) GPTn\_OVF interrupt (n = 0 to 5, 10 to 13)

An interrupt request is generated in the following conditions:

- In saw-wave mode, interrupt requests are enabled at overflows (when the GTCNT counter value changes from GTPR to 0 during up-counting)
- In triangle-wave mode, interrupt requests are enabled at crests (the GTCNT changes from GTPR to GTPR-1)

- In counting by hardware sources, an overflow (GTCNT changes from GTPR to 0 in up count) has occurred.

#### (8) GPTn\_UDF interrupt (n = 0 to 5, 10 to 13)

An interrupt request is generated in the following conditions.

- In saw-wave mode, interrupt requests are enabled at underflows (when the GTCNT counter value changes from 0 to GTPR during down-counting)
- In triangle-wave mode, interrupt requests are enabled at troughs (the GTCNT changes from 0 to 1)
- In counting by hardware sources, underflow (GTCNT changes from 0 to GTPR in down count) has occurred.

About Interrupt signals and interrupt status flags, see [section 21.2.16. GTST : General PWM Timer Status Register](#).

#### (9) GPTn\_PC Interrupt (n = 0 to 3, 10)

When the GTCNT counter value matches with GTADTRA, an interrupt request is generated under the following condition.

- In Up-counting, the interrupt enable bit (ADTRAUEN) in GTINTAD is 1.
- In Down-counting, the interrupt enable bit (ADTRADEN) in GTINTAD is 1. In event count operation performing, this interrupt request isn't generated.

### 21.4.2 DMAC and DTC Activation

The DMAC and DTC can be activated by the interrupt in each channel. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#), [section 16, DMA Controller \(DMAC\)](#), and [section 17, Data Transfer Controller \(DTC\)](#).

## 21.5 A/D Conversion Start Request

The A/D conversion start request can be issued at a compare match between the GTCNT counter and the GTADTRA or GTADTRB register. Up-counting only, down-counting only, or both up-counting and down-counting can be specified by setting the GTINTAD register.

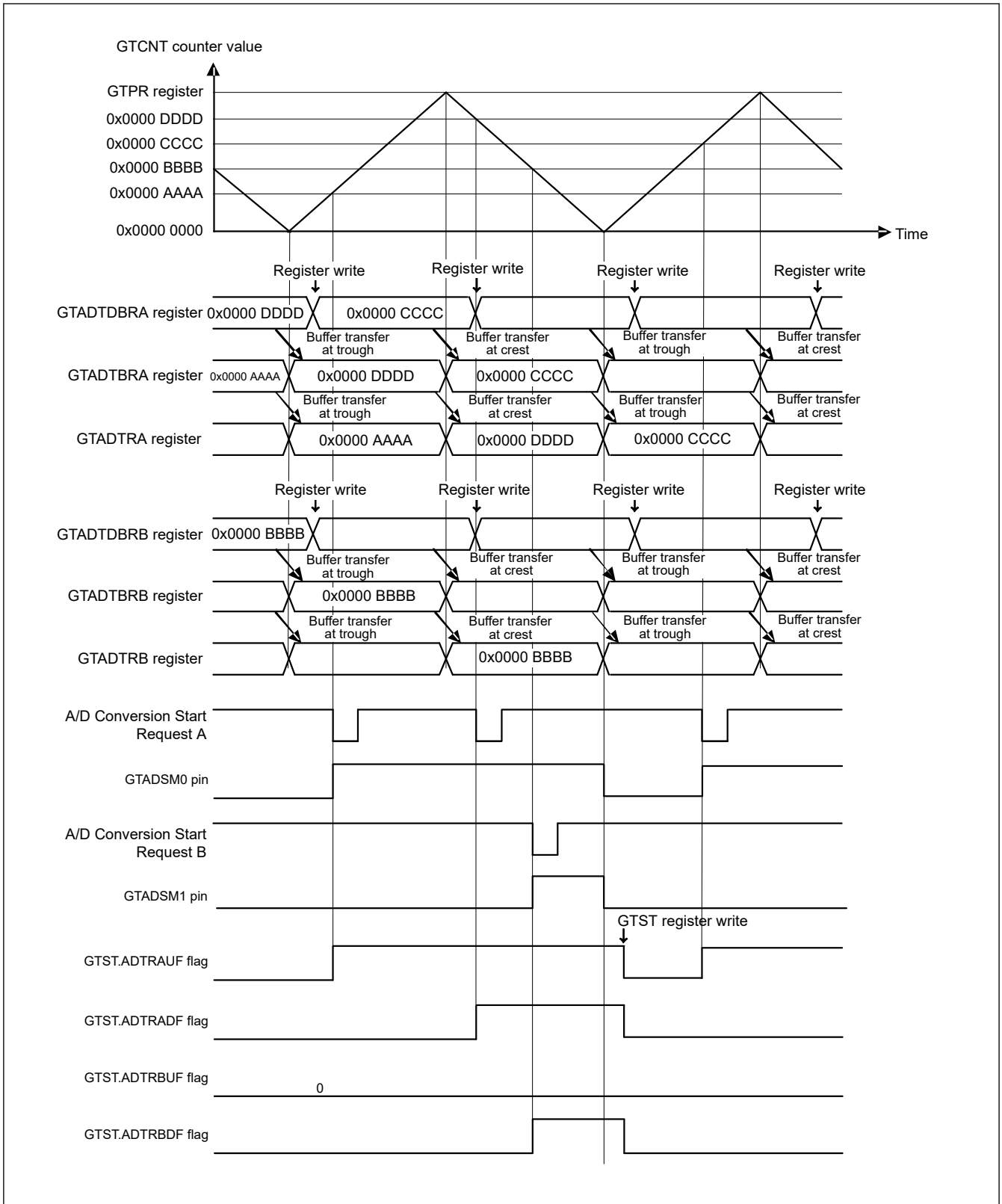
During event count operation, the A/D conversion start request can not be generated.

The A/D conversion start request is directly output to ADC12.

The GTADTRA and GTADTRB registers each has two buffer registers. Buffer operation with the GTADTRA register used together with the GTADTBRA and GTADTDBRA registers, and buffer operation with the GTADTRB register used together with the GTADTBRB and GTADTDBRB registers can be performed.

The timing of the generation of requests to start A/D conversion can be monitored by an external pin. When the A/D conversion start request signal to be monitored is selected in the GTADSMR.ADSMSk bit (k = 0, 1) and when the output is enabled in the ADSMENk bit, a signal is output synchronized with a cycle frame of the timer used to generate the A/D conversion start request signal, of which the output is driven high at the generation of the A/D conversion start request signal by the GTADSMk pin, or at the end of the cycle of which the output is driven low. When a signal to request the start of A/D conversion is generated at the end of the cycle, the generation of this signal has priority in terms of monitoring output and the output remains at the high level till the end of the next cycle. The registers (GTADTRA and GTADTRB) that are sources of generating the A/D conversion start request signals and their counting directions can be checked by the A/D conversion start request flags (ADTRAUF, ADTRADF, ADTRBUF, and ADTRBDF) in the GTST register. When the output of the same A/D conversion start request signal monitoring output is enabled for multiple channels, ORed signals will be output from the GPT32.

[Figure 21.64](#) shows an example of A/D conversion start request operation, [Table 21.41](#) shows example for setting A/D conversion start request operation.



**Figure 21.64 Example of A/D Conversion Start Request Timing Operation (Triangle Waves, Double Buffer Operation, Buffer Transfer at Both Troughs and Crests, A/D Conversion Start Request by GTADTRA Register at Both Up-Counting and Down-Counting, A/D Conversion Start Request by GTADTRB Register at Down-Counting)**

**Table 21.41 Example for Setting A/D Conversion Start Request Timing Operation**

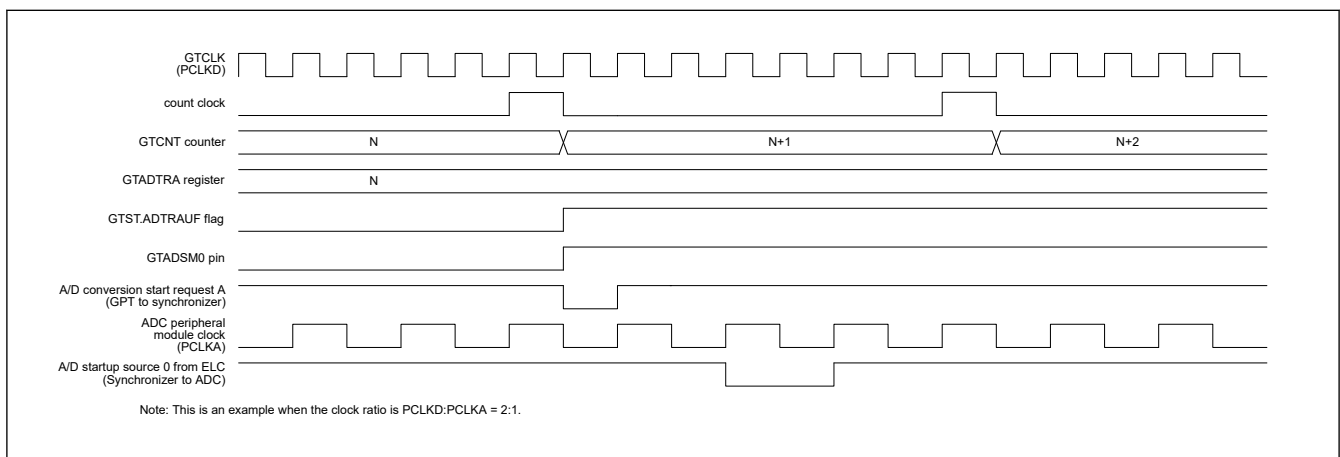
No.	Step Name	Description
1	Set operating mode	Set the operating mode with the GTCR.MD[2:0] bits. In <a href="#">Figure 21.64</a> , 100b, 101b, or 110b (triangle-wave PWM mode) is set.
2	Select count clock	Select the count clock with the GTCR.TPCS[3:0] bits.
3	Set cycle	Set the cycle in the GTPR register.
4	Set initial value for counter	Set the initial value in the GTCNT counter.
5	Set buffer operation	Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in the GTER register. In <a href="#">Figure 21.64</a> , ADTTA[1:0] = 11b, ADTTB[1:0] = 11b, ADTDA = 1, and ADTDB = 1.
6	Set compare match value	Set the A/D conversion start request point in the GTADTRA and GTADTRB registers.
7	Set buffer value	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.
8	Enable A/D conversion start request	Set to enable A/D conversion start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register. In <a href="#">Figure 21.64</a> , ADTRAUEN = 1, ADTRADEN = 1, ADTRBUEN = 0, and ADTRBDEN = 1.
9	Start count operation	Set the GTCR.CST bit to 1 to start count operation.
10	Set buffer value for each cycle	For buffer operation, set the A/D conversion start request point in one cycle after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D conversion start request point in two cycles after the current cycle (in saw-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

[Figure 21.65](#) shows an example for A/D conversion start request timing operation.

This shows an example of the output of A/D conversion start request A for the A/D converter. The A/D conversion start request A signal is output in response to a match in comparison with the GTADTRA register.

If GPT is operating with PCLKD and ADC is operating with PCLKA, A/D conversion start request A is passed to synchronizer on the next rising edge of PCLKA.

A/D conversion start request A is synchronized and passed to ADC.



**Figure 21.65 Example of A/D Conversion Start Request Timing Operation**



## 21.6 Operations Linked by ELC

### 21.6.1 Event Signal Output to ELC

The GPT can perform operation linked with another module set in advance when its interrupt request signal is used as an event signal by the Event Link Controller (ELC).

The A/D conversion start request during up-counting/down-counting can be enabled/disabled individually with the A/D conversion start request enable bit to output events output to ELC.

The GPT has the following ELC event signals:

- Generation of compare match and input capture A interrupt (GPTn\_CCMPA)
- Generation of compare match and input capture B interrupt (GPTn\_CCMPB)
- Generation of compare match C interrupt (GPTn\_CMPC)
- Generation of compare match D interrupt (GPTn\_CMPD)
- Generation of compare match E interrupt (GPTn\_CMPE)
- Generation of compare match F interrupt (GPTn\_CMPF)
- Generation of overflow interrupt (GPTn\_OVF)
- Generation of underflow interrupt (GPTn\_UDF)
- Finish of period count function (GPTm\_PC)

Note: n = 0 to 5, 10 to 13  
m = 0 to 3, 10

### 21.6.2 Event Signal Inputs from ELC

The GPT can perform the following operations in response to a maximum of 8 events from the ELC:

- Start counting, stop counting, clear counting
- Up-counting, down-counting
- Input capture.

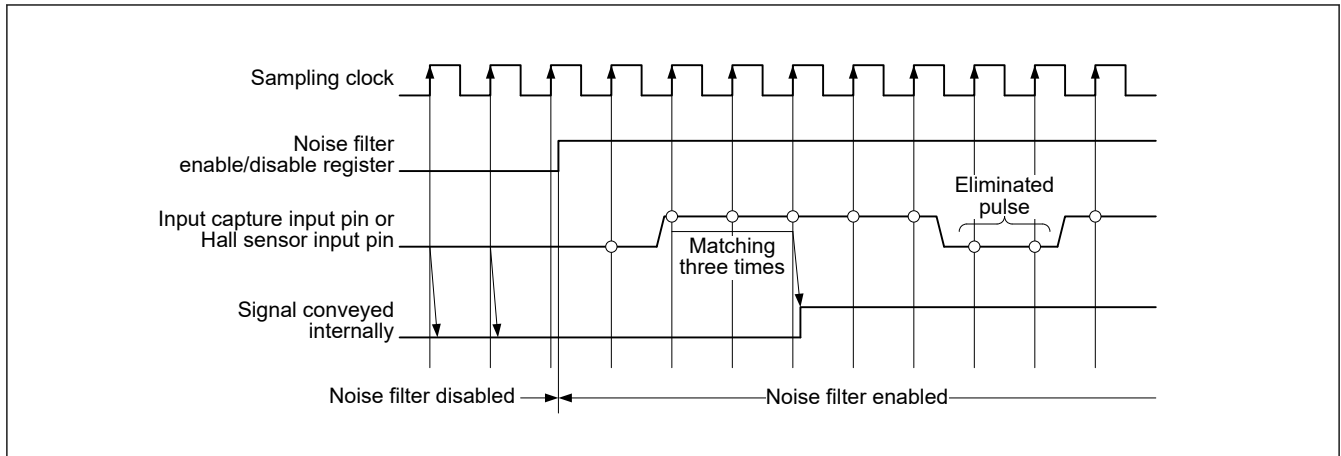
See [section 18, Event Link Controller \(ELC\)](#) for the connection between the ELC and the event signal input.

## 21.7 Noise Filter Function

Each pin for use in input capture and Hall sensor input to the GPT is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses whose length is less than 3 sampling cycles.

The noise filter functionality includes enabling and disabling the noise filter for each pin and setting of the sampling clock for each channel.

[Figure 21.66](#) shows the timing of noise filtering.



**Figure 21.66** Timing of noise filtering

If noise filtering is enabled, the input capture operation or hall sensor input operation is performed on the edges of the noise filtered signal after a delay of  $(\text{sampling interval} \times 2 + \text{PCLKD})$  at the shortest. This is due to the noise filtering for the input capture input or hall sensor input.

## 21.8 Protection Function

### 21.8.1 Write-Protection for Registers

To prevent registers from being accidentally modified, registers can be write-protected in channel units by setting GTWP.WP. Write-protection can be set for the following registers:

GTSSR, GTPSR, GTCSSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTCNT, GTCORA, GTCORB, GTCORC, GTCORD, GTCORE, GTCORF, GTPR, GTPBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTADSMR, GTICLF, GTPC.

Every bit in registers GTSTR, GTSTP and GTCLR which can update the corresponding registers in other channels and can be updated by any of the corresponding registers in other channels conversely, can be protected by setting the GTWP.STRWP, STPWP, and CLRWP bits, respectively, per channel.

Likewise, writing to the GTSECSR and GTSECR registers, which can control all channels by writing to the GTSECSR and GTSECR registers of a given channel, can be enabled or disabled by the setting of the GTWP.CMNWP bit.

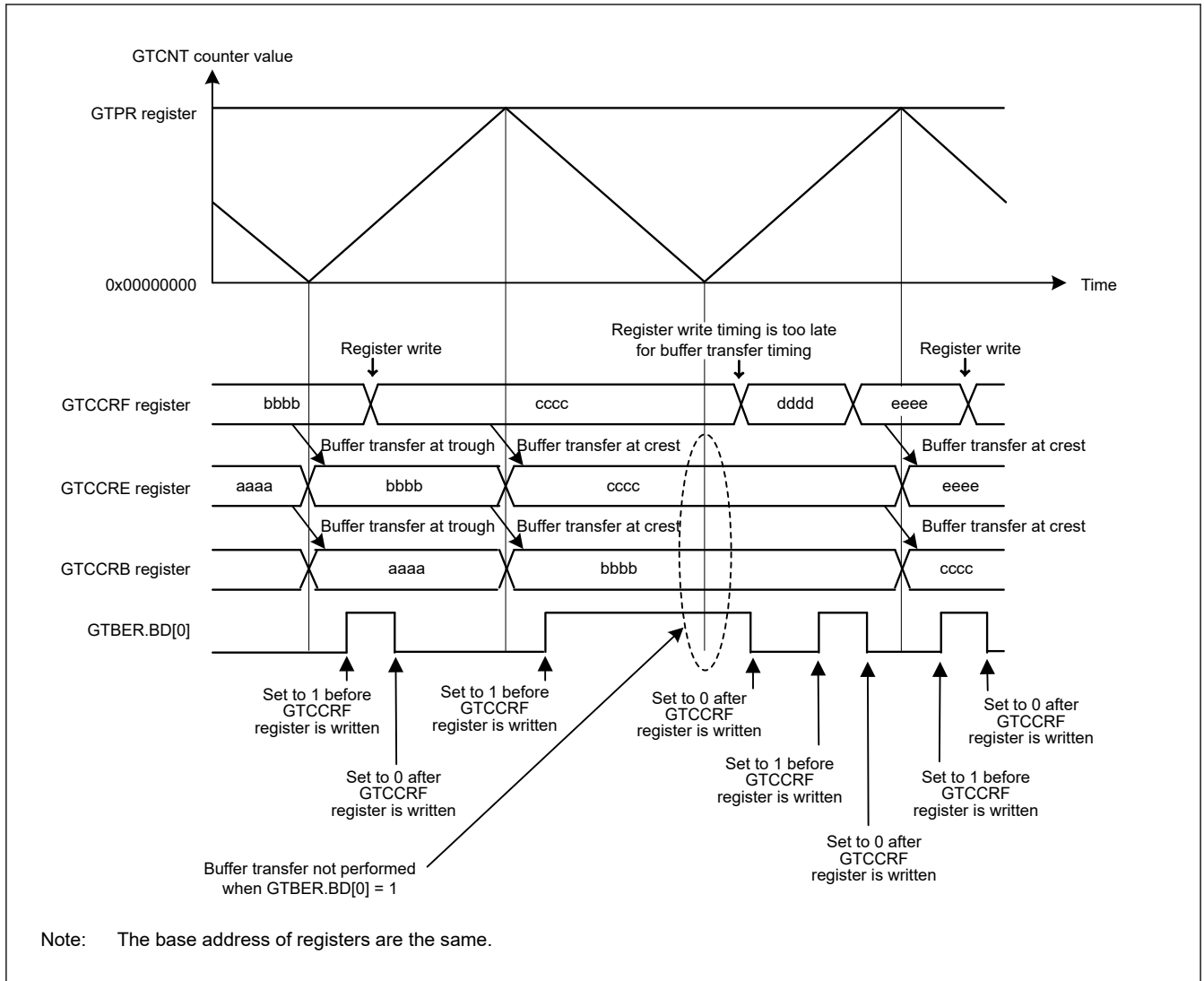
Protection using the GTWP register is only for write operations by the CPU. This protection does not cover updates to registers that occur in association with CPU writes.

### 21.8.2 Disabling of Buffer Operation

If the timing of the buffer register write is delayed relative to the timing for the buffer transfer, buffer operation can be suspended with the GTBER.BD[2], BD[1] and BD[0] bits settings. Specifically, buffer transfer can be temporarily disabled even though a buffer transfer condition is generated during buffer register write, by setting the BD[2], BD[1] and BD[0] bits to 1 (buffer operation disabled) before buffer register write, and setting the bits to 0 (buffer operation enabled) after completion of writing to all the buffer registers.

The BD[2], BD[1] and BD[0] bits can be set on channel basis by writing directly to the GTBER register or it can be set to 0 simultaneously by setting the GTSECR register for multiple channels which were set by the GTSECSR register.

Figure 21.67 shows an example of operation for disabling buffer operation by writing to the GTBER register.



**Figure 21.67 Example of operation for disabling buffer operation with triangle waves, double buffer operation, and buffer transfer at both troughs and crests**

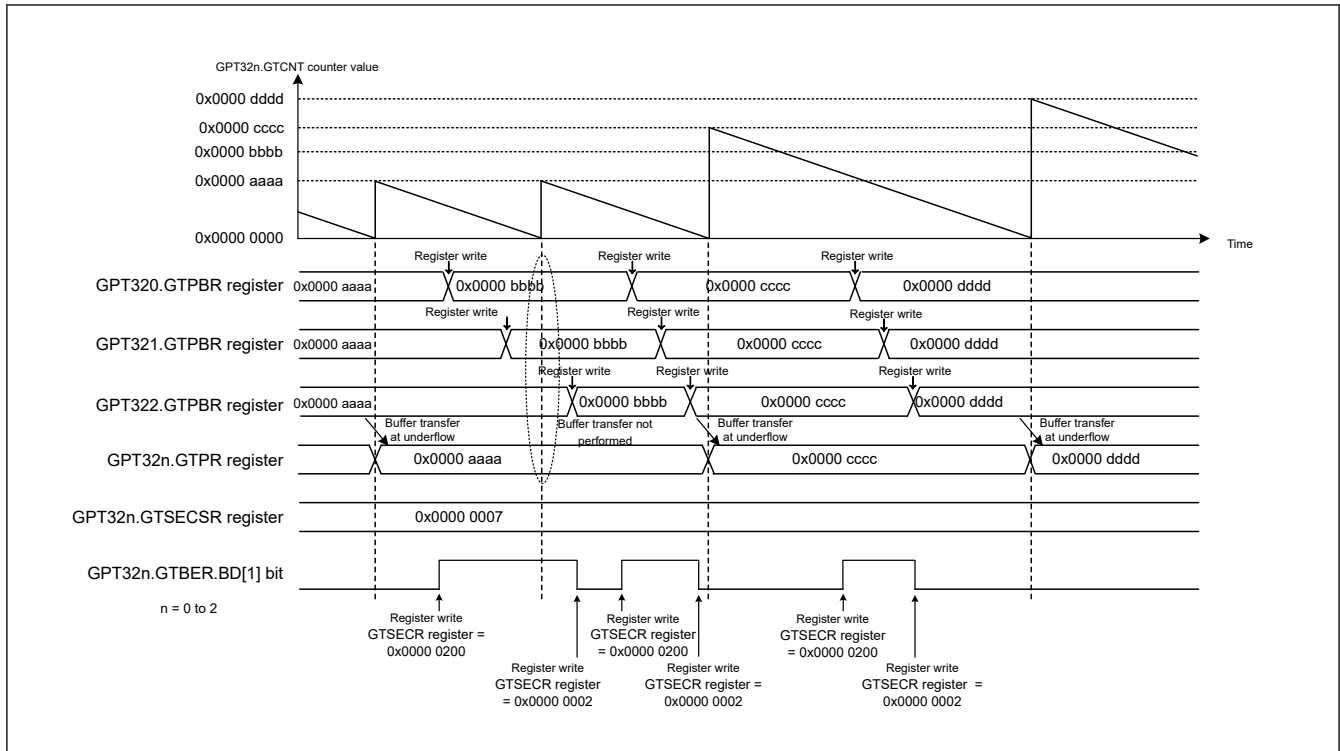
### 21.8.2.1 Simultaneous Control of Buffer Operations of Multiple Channels

The GTBER.BD bit can be set by writing directly to the GTBER register per channel or by making settings in the GTSECR register for multiple channels that have already set in the GTSECSR register.

Follow the procedure below to simultaneously set the GTBER.BD bits of multiple channels.

1. Select the channels for simultaneously setting by the GTSECSR register  
Set the GTSECSR register so that the values at the bit positions for the corresponding channels for simultaneously setting of the GTBER.BD bits become 1. All GTSECSR registers can be updated by writing to the GTSECSR register of any channel.
2. Simultaneously set the GTBER.BD bits by updating the GTSECR register  
In the GTSECR register, set the operation of the GTBER.BD bits (enabling or disabling of buffer operation) which are to be simultaneously set. Writing to a GTSECR register from any channel updates the GTBER.BD bits in all channels corresponding to the bits set as 1 in the GTSECSR register, in accordance with the value of the GTSECR register.

Figure 21.68 show examples of simultaneously controlling the enabling or disabling of buffer operation for multiple channels.



**Figure 21.68 Example of Multiple Channel Operation for Disabling Buffer Operation (Saw Waves, Single Buffer Operation)**

### 21.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 5, 10 to 13, m = A, B)

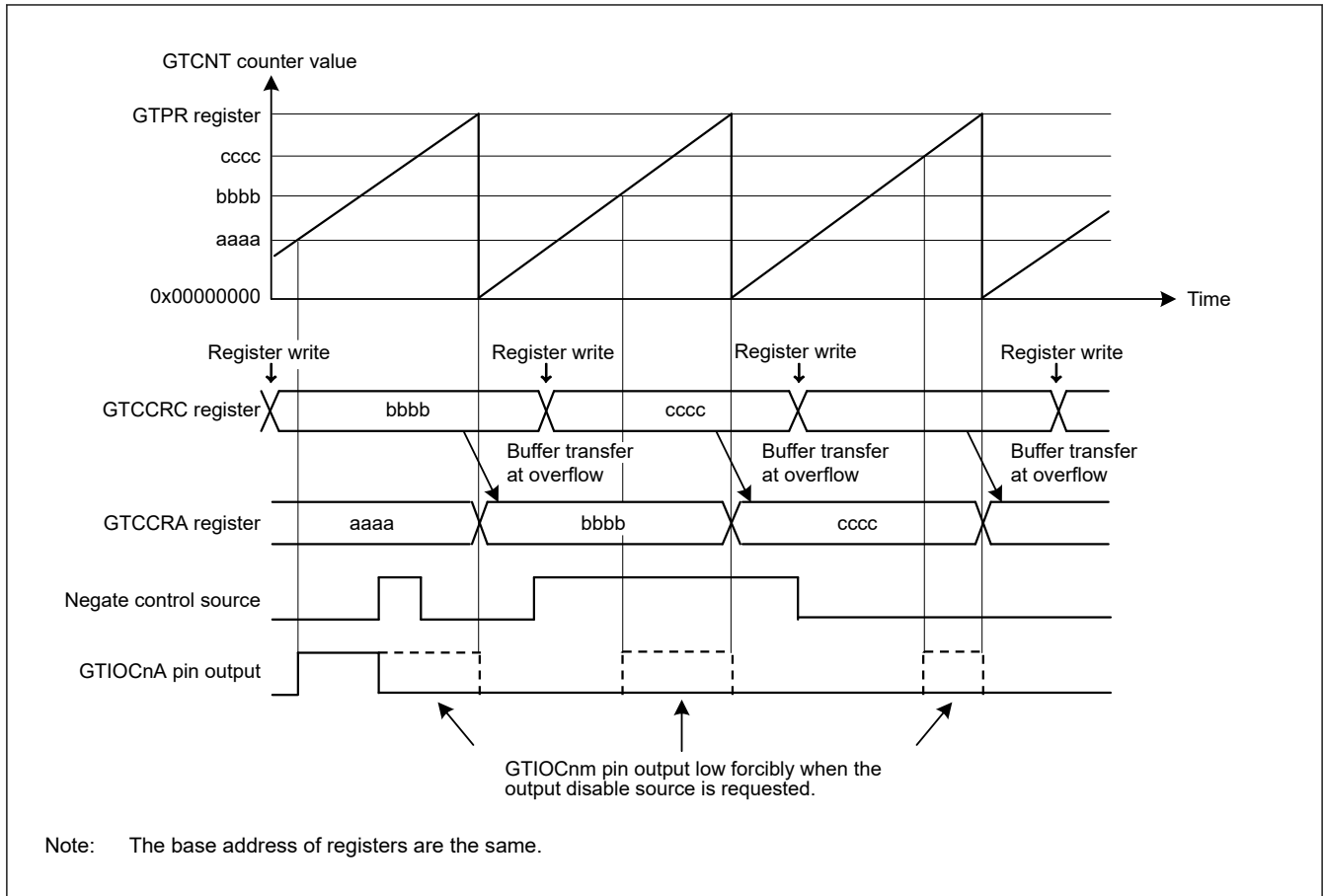
For protection from system failure, the output disable control that changes the GTIOCnm pin output value forcibly is provided for GTIOCnm pin output by the request of output disable from POEG. Output protection is required when the same output level being on the GTIOCnA and GTIOCnB pins is detected. GPT detects this condition and generates output disable requests to POEG according to the setting of the output disable request permission bits, such as GTINTAD.GRPABH, GTINTAD.GRPABL. After the POEG performs the logical OR of the output disable request from each channel and the output disable request from the external input, the POEG generates output disable requests to GPT.

One output disable signal (representing the shared output disable request signal of the GTIOCnA pin and the GTIOCnB pin) out of 4 output disable requests generated by the POEG is selected by setting GTINTAD.GRP[1:0]. The status of the selected disable output request is monitored by reading the GTST.ODF bit. The output level during output disable is set based on the GTIOR.OADF[1:0] bits for the GTIOCnA pin and the GTIOR.OBDF[1:0] setting for the GTIOCnB pin.

The change to the output disable state is performed asynchronously by generating the output disable request from the POEG. The release of the output disable state is performed at end of cycle by terminating the output disable request. It is after 3 PCLKD at shortest when the output disable condition is released after the output disable request becomes no longer satisfied. To reliably control output disabling, clear the flag of POEG for which the condition for the request to disable the output is no longer satisfied after 4 cycles of PCLKD.

When event count is performed or when the output disable state should be released immediately without waiting for end of cycle, GTIOR.OADF[1:0] should be set to 00b (for GTIOCnA pin) or GTIOR.OBDF[1:0] should be set to 00b (for the GTIOCnB pin).

Figure 21.69 shows an example of the GTIOCnm pin output disable control operation. (n = 0 to 5, 10 to 13, m = A, B)



**Figure 21.69 Example of GTIOcNm pin output disable control operation in saw-wave up-counting, buffer operation, active level 1, high output at GTCCRA compare match, low output at cycle end, and low output at output disable (n = 0 to 5, 10 to 13, m = A, B)**

## 21.9 Initialization Method of Output Pins

### 21.9.1 Pin Settings after Reset

The GPT registers are initialized at a reset. Start counting after selecting the port pin function with the PmnPFS register, setting GTIOR.OAE and GTIOR.OBE bits, and outputting the GPT function to external pins.

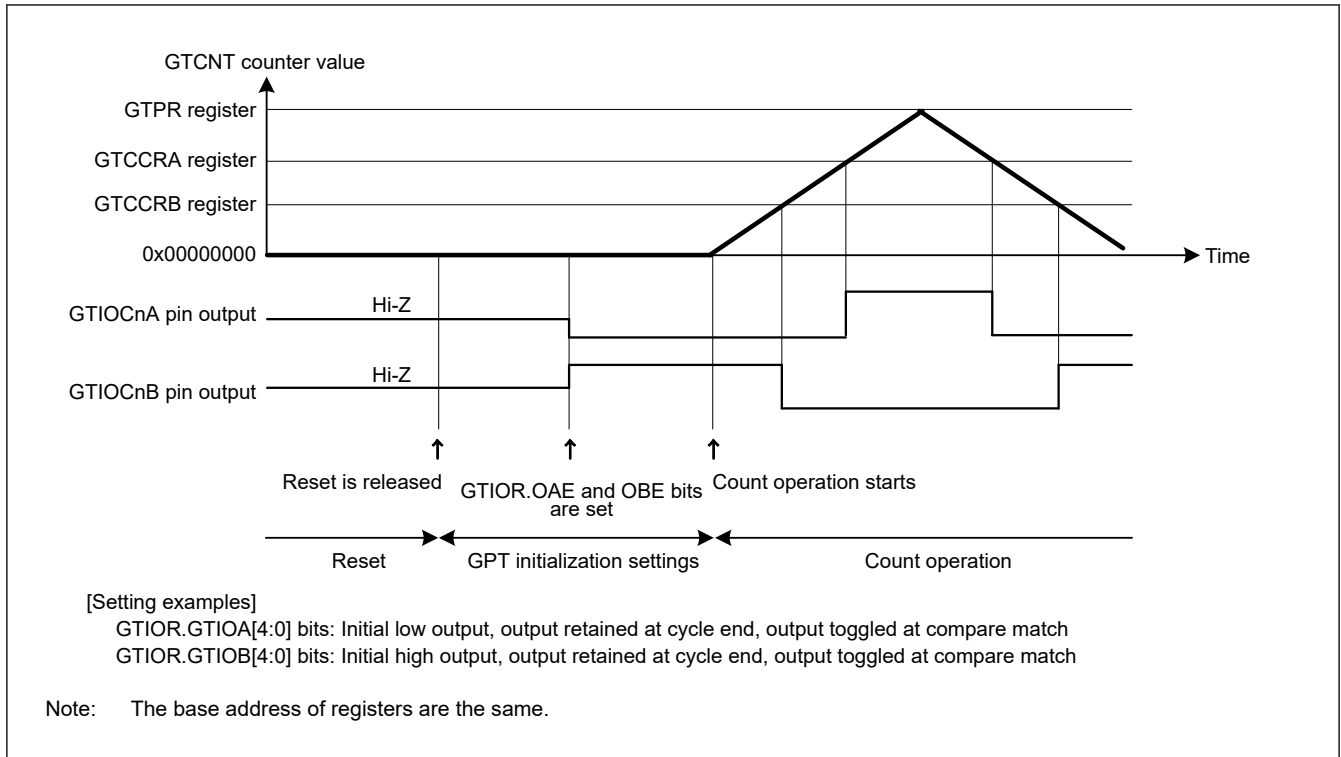


Figure 21.70 Example of pin settings after reset

### 21.9.2 Pin Initialization Due to Error during Operation

If an error occurs during GPT operation, the following four types of pin control can be performed before pin initialization:

- Set the OAHLD and OBHLD bits in GTIOR to 1 and retain the outputs at count stop
- Set the OAHLD and OBHLD bits in GTIOR to 0, specify arbitrary output values at OADFLT and OBDFLT in GTIOR, and output the arbitrary values at count stop
- Set the pin to output an arbitrary value as a general output port by setting the PDR, PODR registers and PmnPFS.PMR bit of the I/O port in advance. Set the OAE and OBE bits in GTIOR to 0, and the control bit associated with the pin in the PMR to 0 to allow arbitrary values to be output from the pin set as a general output port when an error occurs.
- Drive the output to a high impedance state using the POEG function.

If the automatic dead time setting is made, clear the GTDTCR.TDE bit to 0 after counting stops. When counting stops, only the values of registers that are changed by a GPT external source change. If counting is resumed, operation continues from where it stopped. If counting is stopped, the registers must be initialized before counting starts.

## 21.10 Usage Notes

### 21.10.1 Module-Stop Function Setting

The Module Stop Control Register can enable or disable GPT operation. The GPT is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

### 21.10.2 GTCCRn Settings during Compare Match Operation (n = A to F)

#### (1) When automatic dead time setting is made in triangle-wave PWM mode

The GTCCRA register must satisfy all of the following conditions:

- $GTDVU < GTCCRA$
- $0 < GTCCRA < GTPR$

## (2) When automatic dead time setting is not made in triangle-wave PWM mode

The GTCCRA register must be set within the range of  $0 < GTCCRA < GTPR$ . If  $GTCCRA = 0$  or  $GTCCRA = GTPR$  is set, a compare match occurs within the cycle only when  $GTCCRA = 0$  or  $GTCCRA = GTPR$  is satisfied. When  $GTCCRA > GTPR$ , no compare match occurs.

Similarly, GTCCRB must be set within the range of  $0 < GTCCRB < GTPR$ . If  $GTCCRB = 0$  or  $GTCCRB = GTPR$  is set, a compare match occurs within the cycle only when  $GTCCRB = 0$  or  $GTCCRB = GTPR$  is satisfied. When  $GTCCRB > GTPR$ , no compare match occurs.

## (3) When automatic dead time setting is made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, the correct output waveforms with secured dead time may not be obtained.

- In up-counting:  $GTCCRC < GTCCRD$ ,  $GTCCRC > GTDVU$ ,  $GTCCRD < GTPR - GTDVU$
- In down-counting:  $GTCCRC > GTCCRD$ ,  $GTCCRC < GTPR - GTDVU$ ,  $GTCCRD > GTDVU$

## (4) When automatic dead time setting is not made in saw-wave one-shot pulse mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting:  $0 < GTCCRC < GTCCRD < GTPR$
- In down-counting:  $GTPR > GTCCRC > GTCCRD > 0$

Similarly, GTCCRE and GTCCRF must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting:  $0 < GTCCRE < GTCCRF < GTPR$
- In down-counting:  $GTPR > GTCCRE > GTCCRF > 0$ .

## (5) In saw-wave PWM mode

The GTCCRA register must be set with the range of  $0 < GTCCRA < GTPR$ . If  $GTCCRA = 0$  or  $GTCCRA = GTPR$  is set, a compare match occurs within the cycle only when  $GTCCRA = 0$  or  $GTCCRA = GTPR$  is satisfied. If  $GTCCRA > GTPR$  is set, no compare match occurs.

Similarly, GTCCRB must be set with the range of  $0 < GTCCRB < GTPR$ . If  $GTCCRB = 0$  or  $GTCCRB = GTPR$  is set, a compare match occurs within the cycle only when  $GTCCRB = 0$  or  $GTCCRB = GTPR$  is satisfied. If  $GTCCRB > GTPR$  is set, no compare match occurs.

### 21.10.3 Setting Range for GTCNT Counter

The GTCNT counter register must be set with the range of  $0 \leq GTCNT \leq GTPR$ .

### 21.10.4 Starting and Stopping the GTCNT Counter

The control timing of starting and stopping the GTCNT counter by the GTCR.CST bit synchronizes the count clock that is selected in GTCR.TPCS[3:0]. When GTCR.CST is updated, the GTCNT counter starts/stops after a count clock that is selected in GTCR.TPCS[3:0]. Therefore, an event generated before the GTCNT counter actually starts is ignored, resulting in situations in which an event is accepted or an interrupt occurs after GTCR.CST is set to 0.

### 21.10.5 Priority Order of Each Event

#### (1) GTCNT register

Table 21.42 shows a priority order of events updating the GTCNT register.

**Table 21.42 Priority order of sources updating GTCNT**

Source updating GTCNT	Priority order
Writing by CPU (writing to GTCNT/GTCLR)	High
Clear by hardware sources set in GTCR	↑
Count up or down by hardware sources set in GTUPSR/GTDNSR	↑
Count operation	Low

If up-counting and down-counting by hardware sources occur at the same time, the GTCNT counter value does not change. When there is a conflict between updating the GTCNT register and reading by the CPU, pre-update data is read.

### (2) GTCR.CST bit

When there is a conflict between starting/stopping by hardware sources set in the GTSSR/GTPSR registers and writing by the CPU (writing to GTCR/GTSTR/GTSTP registers), the writing by CPU has priority over the starting/stopping by hardware sources.

In case that stop by the period count function conflicts with start by the CPU writing (GTCR register writing/GTSTR register writing), the period count function is finished with setting the GTST.PCF flag. The CST bit is not changed and the GTCNT continues to count.

When there is a conflict between starting by hardware sources set in the GTSSR register and stopping by hardware sources set in GTPSR register, the GTCR.CST bit value does not change. When there is a conflict between updating the GTCR.CST bit and reading by the CPU (reading from GTCR/GTSTR/GTSTP registers), pre-update data is read.

### (3) GTCCRm registers (m = A to F)

When there is a conflict between input capture/buffer transfer operation and writing to the GTCCRm registers, the writing to GTCCRm registers has priority over input capture/buffer transfer operation. When there is a conflict between input capture and writing to the counter register by the CPU or updating the counter register by hardware sources, the pre-update counter value is captured. When there is a conflict between updating the GTCCRm registers and reading by the CPU, pre-update data is read.

### (4) GTPR register

When there is a conflict between buffer transfer operation and writing to the GTPR register, writing to GTPR register has priority over buffer transfer operation. When there is a conflict between updating GTPR register and reading by the CPU, pre-update data is read.

### (5) GTADTRm registers (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTADTRm register, writing to GTADTRm register has priority over buffer transfer operation.

When there is a conflict between updating the GTADTRm register and reading by the CPU, pre-update data is read.

### (6) GTDVU registers

When there is a conflict between buffer transfer operation and writing to GTDVU register, writing to GTDVU register has priority over buffer transfer operation.

When there is a conflict between updating the GTDVU register and reading by the CPU, pre-update data is read.

### (7) GTIOR.GTIOm registers (m = A, B)

When there is a conflict between buffer transfer operation and writing to GTIOR.GTIOm register, writing to GTIOR.GTIOm register has priority over buffer transfer operation.

When there is a conflict between updating the GTIOR.GTIOm and reading by the CPU, pre-update data is read.

## 21.10.6 Prohibit Invalid Register Setting

The register settings that are instructed to be invalid, such as "The setting is invalid during event counting operation", are not guaranteed. These setting is prohibited.



## 22. Low Power Asynchronous General Purpose Timer (AGT)

### 22.1 Overview

The Low Power Asynchronous General Purpose Timer (AGT) is a 16-bit timer that can be used for pulse output, external pulse width or period measurement, and counting external events. This timer consists of a reload register and a down counter. The reload register and the down counter are allocated to the same address, and can be accessed with the AGT register.

Table 22.1 lists the AGT specifications, Figure 22.1 shows a block diagram, and Table 22.2 lists the I/O pins.

**Table 22.1 AGT specifications**

Parameter		Description
Operating modes	Timer mode	The count source is counted
	Pulse output mode	The count source is counted and the output is inverted at each timer underflow
	Event counter mode	An external event is counted
	Pulse width measurement mode	An external pulse width is measured
	Pulse period measurement mode	An external pulse period is measured
Number of Channels		16 bits × 2 channels (AGTn (n = 0, 1))
Count source (operating clock) <sup>2</sup>	Timer mode	PCLKB, PCLKB/2, PCLKB/8, AGTLCLK/d, AGTSCLK/d (d = 1, 2, 4, 8, 16, 32, 64, or 128), or underflow signal of AGT0 selectable.*1
	Pulse output mode	
	Pulse width measurement mode	
	Pulse period measurement mode	
	Event counting mode	External event input
Interrupt and Event Link function		<ul style="list-style-type: none"> <li>• Underflow event signal or measurement complete event signal <ul style="list-style-type: none"> <li>– When the counter underflows</li> <li>– When the measurement of the active width of the external input pin (AGTIOn) completes in pulse width measurement mode</li> <li>– When the set edge of the external input pin (AGTIOn) is input in pulse period measurement mode.</li> </ul> </li> <li>• Compare match A event signal <ul style="list-style-type: none"> <li>– When the values of AGT register and AGTCMA register matched (compare match A function enabled).</li> </ul> </li> <li>• Compare match B event signal <ul style="list-style-type: none"> <li>– When the values of AGT and AGTCMB registers matched (compare match B function enabled).</li> </ul> </li> <li>• Return from Software Standby mode can be performed with AGT1_AGTI, AGT1_AGTCMAI, or AGT1_AGTCMBI*3</li> </ul>
Selectable functions		<ul style="list-style-type: none"> <li>• Compare match function One or two of the AGT Compare Match A register and AGT Compare Match B register is selectable.</li> </ul>
Module-stop function		Module-stop state can be set for each channel to reduce power consumption.
TrustZone Filter		Security and Privilege attribution can be set for each channels.

Note 1. AGT0 cannot use underflow signal. AGT1 connects directly with the underflow event signal from the AGT0 timer.

Note 2. Satisfy the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source clock.

Note 3. For details, see section 10, Low Power Modes.

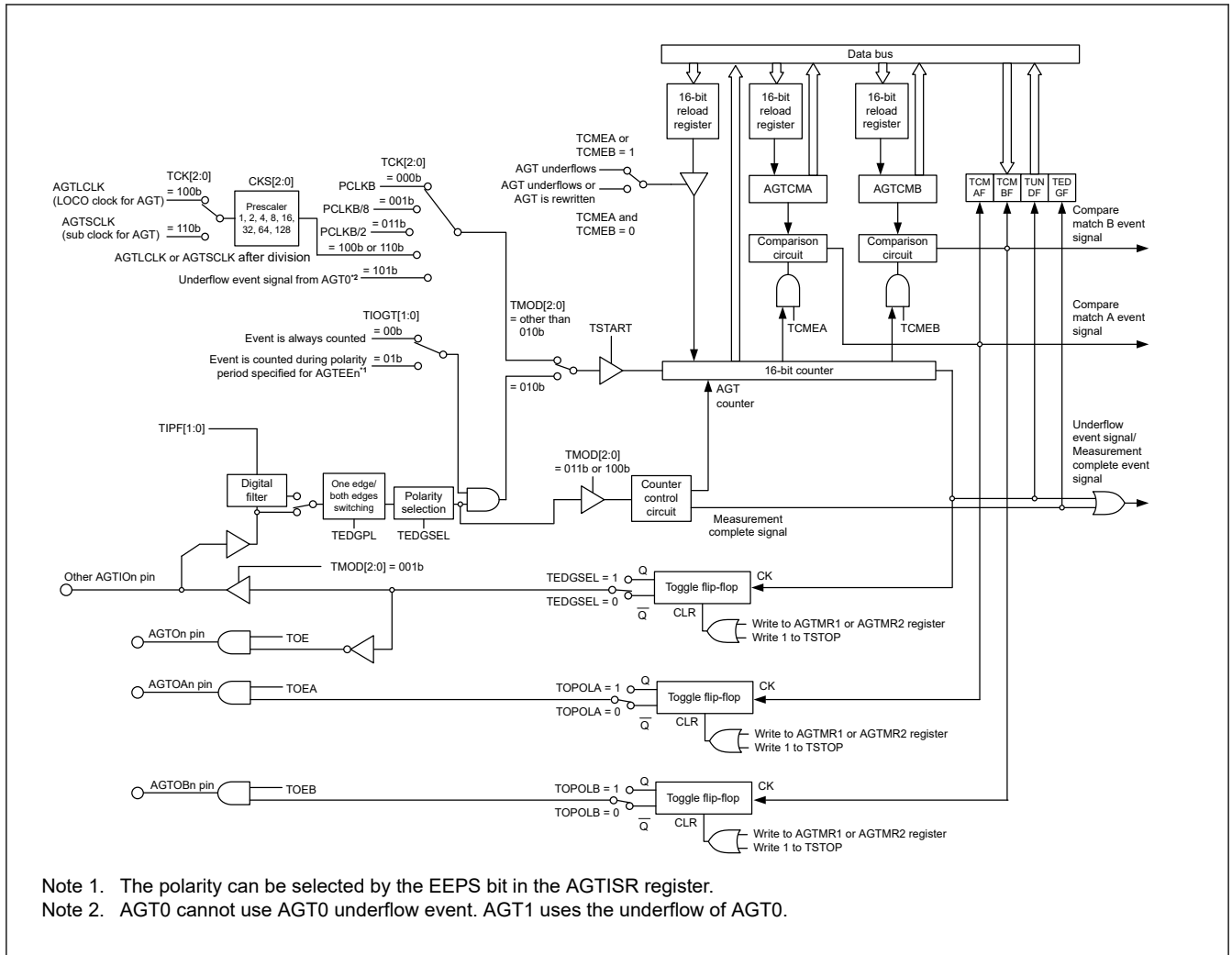


Figure 22.1 AGT block diagram

Table 22.2 AGT I/O pins

Pin name	I/O	Function
AGTEEn	Input	External event input enable for AGT
AGTIOn	Input/output	External event input and pulse output for AGT
AGTOn	Output	Pulse output for AGT
AGTOAn	Output	Compare match A output for AGT
AGTOBn	Output	Compare match B output for AGT

Note: Channel number: n = 0, 1

## 22.2 Register Descriptions

### 22.2.1 AGT : AGT Counter Register

Base address:  $AGTn = 0x4022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $AGTn\_NS = 0x5022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	[Empty box for bit field]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	16-bit counter and reload register Setting range : 0x0000 to 0xFFFF	R/W

Note: S-TYPE-3, P-TYPE-3

AGTn.AGT is a 16-bit register. The write value is written to the reload register and the read value is read from the counter.

The states of the reload register and the counter change according to the TSTART bit in the AGTCR register and TCMEA/TCMEB bit in the AGTCMSR register. For details, see [section 22.3.1. Reload Register and Counter Rewrite Operation](#).

When 1 is written to the TSTOP bit in the AGTCR register, AGT counter is forcibly stopped and set to 0xFFFF.

When the TCK[2:0] bits setting in the AGTMR1 register are a value other than 001b (PCLKB/8) or 011b (PCLKB/2), if the AGT register is set to 0x0000, a request signal to the ICU, the DTC, the DMAC, and the ELC is generated once immediately after the count starts. The AGTOn, AGTIO pin output are toggled.

When the AGT register is set to 0x0000 in event counter mode, regardless of the value of TCK[2:0] bits, a request signal to the ICU, the DTC, the DMAC, and the ELC is generated once immediately after the count starts.

In addition, the AGTOn pin output is toggled even during a period other than the specified count period. When the AGT register is set to 0x0001 or more, a request signal is generated each time AGT underflows.

### 22.2.2 AGTCMA : AGT Compare Match A Register

Base address:  $AGTn = 0x4022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $AGTn\_NS = 0x5022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	[Empty box for bit field]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	16-bit compare match A data is stored.*1 Setting range : 0x0000 to 0xFFFF	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Set the AGTCMA register to 0xFFFF when compare match A is not used.

The AGTCMA register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register A change according to the TSTART bit in the AGTCR register. For details, see [section 22.3.2. Reload Register and AGT Compare Match A/B Register Rewrite Operation](#).

### 22.2.3 AGTCMB : AGT Compare Match B Register

Base address:  $AGT_n = 0x4022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $AGT\_n\_NS = 0x5022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x04

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

--

Value after reset: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
15:0	n/a	16-bit compare match B data is stored.*1 Setting range : 0x0000 to 0xFFFF	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Set the AGTCMB register to 0xFFFF when compare match B is not used.

The AGTCMB register is a read/write register to set a value for compare match with the AGT counter. The states of the reload register and compare register B change according to the TSTART bit in the AGTCR register. For details, see [section 22.3.2. Reload Register and AGT Compare Match A/B Register Rewrite Operation](#).

### 22.2.4 AGTCR : AGT Control Register

Base address:  $AGT_n = 0x4022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $AGT\_n\_NS = 0x5022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x08

Bit position: 7 6 5 4 3 2 1 0

TCMB	TCMA	TUNDF	TEDGF	—	TSTO	TCST	TSTART
F	F	F	F		P	F	RT

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	TSTART	AGT Count Start*2 0: Count stops 1: Count starts	R/W
1	TCSTF	AGT Count Status Flag*2 0: Count stopped 1: Count in progress	R
2	TSTOP	AGT Count Forced Stop*1 0: Writing is invalid 1: The count is forcibly stopped	W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TEDGF	Active Edge Judgment Flag 0: No active edge received 1: Active edge received	R/(W)*3
5	TUNDF	Underflow Flag 0: No underflow 1: Underflow	R/(W)*3
6	TCMAF	Compare Match A Flag 0: No match 1: Match	R/(W)*3
7	TCMBF	Compare Match B Flag 0: No match 1: Match	R/(W)*3

Note: S-TYPE-3, P-TYPE-3

Note 1. When 1 (count is forcibly stopped) is written to the TSTOP bit, the TSTART bit and TCSTF flag are initialized at the same time. The pulse output level is also initialized. The read value is 0.

Note 2. For information on using the TSTART bit and TCSTF flag, see [section 22.4.1. Count Operation Start and Stop Control](#).

Note 3. Only 0 can be written to clear the flag.

### **TSTART bit (AGT Count Start)**

The count operation is started by writing 1 to the TSTART bit and stopped by writing 0. When the TSTART bit is set to 1 (count starts), the TCSTF flag is set to 1 (count in progress) in synchronization with the count source. Also, after 0 is written to the TSTART bit, the TCSTF flag is set to 0 (count stops) in synchronization with the count source. For details, see [section 22.4.1. Count Operation Start and Stop Control](#).

### **TCSTF flag (AGT Count Status Flag)**

The TCSTF flag indicates the AGT count status.

[Setting condition]

- When 1 is written to the TSTART bit (the TCSTF flag is set to 1 in synchronization with the count source).

[Clearing conditions]

- When 0 is written to the TSTART bit (the TCSTF flag is set to 0 in synchronization with the count source)
- When 1 is written to the TSTOP bit.

### **TSTOP bit (AGT Count Forced Stop)**

When 1 is written to the TSTOP bit, the count is forcibly stopped. The read value is 0.

### **TEDGF flag (Active Edge Judgment Flag)**

The TEDGF flag indicates that an active edge was detected.

[Setting condition]

- When the measurement of the active width of the external input pin (AGTIO<sub>n</sub>) is complete in pulse width measurement mode
- When the set edge of the external input pin (AGTIO<sub>n</sub>) is input in pulse period measurement mode.

[Clearing condition]

- When 0 is written to this flag by software.

### **TUNDF flag (Underflow Flag)**

The TUNDF flag indicates that the counter underflowed.

[Setting condition]

- When the counter underflows.

[Clearing condition]

- When 0 is written to this flag by software.

### **TCMAF flag (Compare Match A Flag)**

The TCMAF flag indicates that compare match A was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMA register.

[Clearing condition]

- When 0 is written to this flag by software.

### **TCMBF flag (Compare Match B Flag)**

The TCMBF flag indicates that compare match B was detected.

[Setting condition]

- When the value in the AGT register matches the value in the AGTCMB register.

[Clearing condition]

- When 0 is written to this flag by software.

### 22.2.5 AGTMR1 : AGT Mode Register 1

Base address: AGTn = 0x4022\_1000 + 0x0100 × n (n = 0, 1)  
AGTn\_NS = 0x5022\_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TCK[2:0]			TEDG PL	TMOD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TMOD[2:0]	Operating Mode*3 0 0 0: Timer mode 0 0 1: Pulse output mode 0 1 0: Event counter mode 0 1 1: Pulse width measurement mode 1 0 0: Pulse period measurement mode Others: Setting prohibited	R/W
3	TEDGPL	Edge Polarity*4 0: Single-edge 1: Both-edge	R/W
6:4	TCK[2:0]	Count Source*1 *2 *5 *7 0 0 0: PCLKB 0 0 1: PCLKB/8 0 1 1: PCLKB/2 1 0 0: Divided clock AGTLCLK specified by CKS[2:0] bits in the AGTMR2 register 1 0 1: Underflow event signal from AGT0*6 1 1 0: Divided clock AGTSCLK specified by CKS[2:0] bits in the AGTMR2 register Others: Setting prohibited	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Write access to the AGTMR1 register initializes the output from the AGTOn, AGTIOOn, AGTOAn, and AGTOBn pins. For details on the output level at initialization, see [section 22.2.7. AGTIOC : AGT I/O Control Register](#).

Note 1. When event counter mode is selected, the external input pin (AGTIOOn) is selected as the count source regardless of the setting of TCK[2:0] bits.

Note 2. Do not switch count sources during count operation. Only switch count sources when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).

Note 3. The operating mode can only be changed when the count is stopped while both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count is stopped). Do not change the operating mode during count operation.

Note 4. The TEDGPL bit is enabled only in event counter mode.

Note 5. To run AGT in Software Standby mode, select AGTLCLK or AGTSCLK (TCK[2:0] = 100b, 110b).

Note 6. AGT0 cannot use AGT0 underflow (setting prohibited). AGT1 uses the AGT0 underflow.

Note 7. Do not change the TCK[2:0] bits when the CKS[2:0] bits in the AGTMR2 register is not 000b. First, change the CKS[2:0] bits in the AGTMR2 register to 000b. Then change the TCK[2:0] bits and wait for one cycle of the count source.

## 22.2.6 AGTMR2 : AGT Mode Register 2

Base address: AGTn = 0x4022\_1000 + 0x0100 × n (n = 0, 1)  
AGTn\_NS = 0x5022\_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LPM	—	—	—	—	CKS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CKS[2:0]	AGTLCLK or AGTSCLK Count Source Clock Frequency Division Ratio*1 *2 *3 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	LPM	Low Power Mode 0: Normal mode 1: Low power mode	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Do not rewrite the CKS[2:0] bits during count operation. Only rewrite the CKS[2:0] bits when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).

Note 2. When count source is AGTLCLK or AGTSCLK, the switch of CKS[2:0] bits is valid.

Note 3. Do not switch the TCK[2:0] bits in the AGTMR1 register when CKS[2:0] bits are not 000b. Switch the TCK[2:0] bits in the AGTMR1 register after CKS[2:0] bits are set to 000b, and wait for 1 cycle of the count source.

### CKS[2:0] bit (AGTLCLK or AGTSCLK Count Source Clock Frequency Division Ratio)

CKS[2:0] bits select the Count Source Clock Frequency Division Ratio for AGTLCLK or AGTSCLK.

### LPM bit (Low Power Mode)

The LPM bit sets the low power operation, which impacts access to certain AGT registers. Set this bit to 1 to operate in low power.

When this bit is 1, access to the following registers is prohibited:

- AGT/AGTCMA/AGTCMB/AGTCR.

After this bit is switched from 1 to 0, the first access to the register is constrained as follows:

- When reading from the AGT register, read AGT register twice. Only the second reading of data is valid.
- When writing to the AGT, AGTCMA, AGTCMB, and AGTCR register, allow at least 2 cycles of the count source clock when writing to the register.
- When confirm the value written to the AGT, AGTCMA, AGTCMB, and AGTCR registers.
  - When the count operation is stopped; after writing data, it can be read in the next cycle.
  - When the count operation is operating; after writing data, it can be read 4 cycles after the count source clock.

Figure 22.2 shows the flow of how to write LPM bit

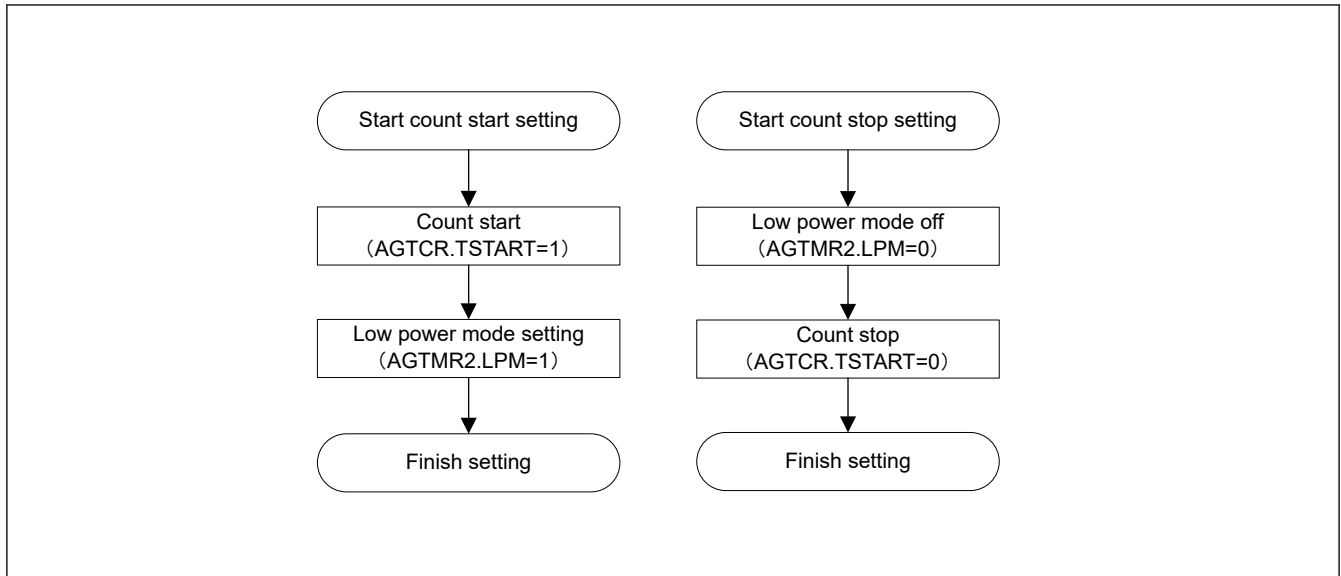


Figure 22.2 LPM how to write flow chart

### 22.2.7 AGTIOC : AGT I/O Control Register

Base address: AGTn = 0x4022\_1000 + 0x0100 × n (n = 0, 1)  
 AGTn\_NS = 0x5022\_1000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIOGT[1:0]		TIPF[1:0]		—	TOE	—	TEDGSEL

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	TEDGSEL	I/O Polarity Switch Function varies depending on the operating mode (see <a href="#">Table 22.3</a> and <a href="#">Table 22.4</a> ).	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	TOE	AGTOn pin Output Enable 0: AGTOn pin output disabled 1: AGTOn pin output enabled	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TIPF[1:0]	Input Filter*3 These bits specifies the sampling frequency of the filter for the AGTIOOn input. If the input to the AGTIOOn pin is sampled and the value matches three successive times, that value is taken as the input value. 0 0: No filter 0 1: Filter sampled at PCLKB 1 0: Filter sampled at PCLKB/8 1 1: Filter sampled at PCLKB/32	R/W
7:6	TIOGT[1:0]	Count Control*1 *2 0 0: Event is always counted 0 1: Event is counted during polarity period specified for AGTEEn pin Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. When AGTEEn pin is used, the polarity to count an event can be selected with the EEPS bit in the AGTISR register.

Note 2. TIOGT[1:0] bits are enabled only in event counter mode.

Note 3. When event counter mode operation is performed during Software Standby mode, the digital filter function cannot be used.

#### TEDGSEL bit (I/O Polarity Switch)

The TEDGSEL bit switches the AGTOn pin output polarity and the AGTIOOn pin input/output edge and polarity.



In pulse output mode, it only controls polarity of the AGTOn pin output and AGTIO pin output. AGTOn pin output and AGTIO pin output are initialized when the AGTMR1 register is written or the TSTOP bit in the AGTCR register is written with 1.

### TOE bit (AGTOn pin Output Enable)

The TOE bit selects whether the AGTOn pin output is disabled or enabled.

### TIPF[1:0] bits (Input Filter)

The TIPF[1:0] bits specify the sampling frequency of the AGTIO pin input filter. When the input to the AGTIO pin is sampled and the values match three times in succession, the value is regarded as the input value.

### TIOGT[1:0] bits (Count Control)

The TIOGT[1:0] bits control the event count.

**Table 22.3 AGTIO pin I/O edge and polarity switching**

Operating mode	Function
Timer mode	Not used
Pulse output mode	0: Output is started at high (initialization level: high) i.e. inverted output 1: Output is started at low (initialization level: low). i.e. normal output
Event counter mode	0: Count on rising edge 1: Count on falling edge.
Pulse width measurement mode	0: Low-level width is measured 1: High-level width is measured.
Pulse period measurement mode	0: Measure from one rising edge to the next rising edge 1: Measure from one falling edge to the next falling edge.

**Table 22.4 AGTOn pin output polarity switching**

Operating mode	Function
All modes	0: Output is started at low (initial level: low): Normal output 1: Output is started at high (initial level: high): Inverted output

## 22.2.8 AGTISR : AGT Event Pin Select Register

Base address:  $AGTn = 0x4022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $AGTn\_NS = 0x5022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	EEPS	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	EEPS	AGTEEn Polarity Selection 0: An event is counted during the low-level period 1: An event is counted during the high-level period	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

### EEPS bit (AGTEEn Polarity Selection)

The EEPS bit selects the polarity of events to be counted.

### 22.2.9 AGTCMSR : AGT Compare Match Function Select Register

Base address:  $AGTn = 0x4022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $AGTn\_NS = 0x5022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x0E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TOPO LB	TOEB	TCME B	—	TOPO LA	TOEA	TCME A
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCMEA	AGT Compare Match A Register Enable*1 *2 0: AGT Compare match A register disabled 1: AGT Compare match A register enabled	R/W
1	TOEA	AGTOAn Pin Output Enable*1 *2 0: AGTOAn pin output disabled 1: AGTOAn pin output enabled	R/W
2	TOPOLA	AGTOAn Pin Polarity Select*1 *2 0: AGTOAn pin output is started on low. i.e. normal output 1: AGTOAn pin output is started on high. i.e. inverted output	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TCMEB	AGT Compare Match B Register Enable*1 *2 0: Compare match B register disabled 1: Compare match B register enabled	R/W
5	TOEB	AGTOBn Pin Output Enable*1 *2 0: AGTOBn pin output disabled 1: AGTOBn pin output enabled	R/W
6	TOPOLB	AGTOBn Pin Polarity Select*1 *2 0: AGTOBn pin output is started on low. i.e. normal output 1: AGTOBn pin output is started on high. i.e. inverted output	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Do not rewrite the AGTCMSR register during a count operation. Only rewrite the AGTCMSR register when both the TSTART bit and TCSTF flag in the AGTCR register are set to 0 (count stops).

Note 2. Do not set 1 when in pulse width measurement mode or pulse period measurement mode.

### 22.2.10 AGTIOSEL : AGT Pin Select Register

Base address:  $AGTn = 0x4022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $AGTn\_NS = 0x5022\_1000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x0F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TIES	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	TIES	AGTIOn Pin Input Enable 0: External event input is disabled during Software Standby mode 1: External event input is enabled during Software Standby mode	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The AGTIOSEL register sets the AGTIO pin when using the AGTIO pin in Software Standby mode.

**TIES bit (AGTIO Pin Input Enable)**

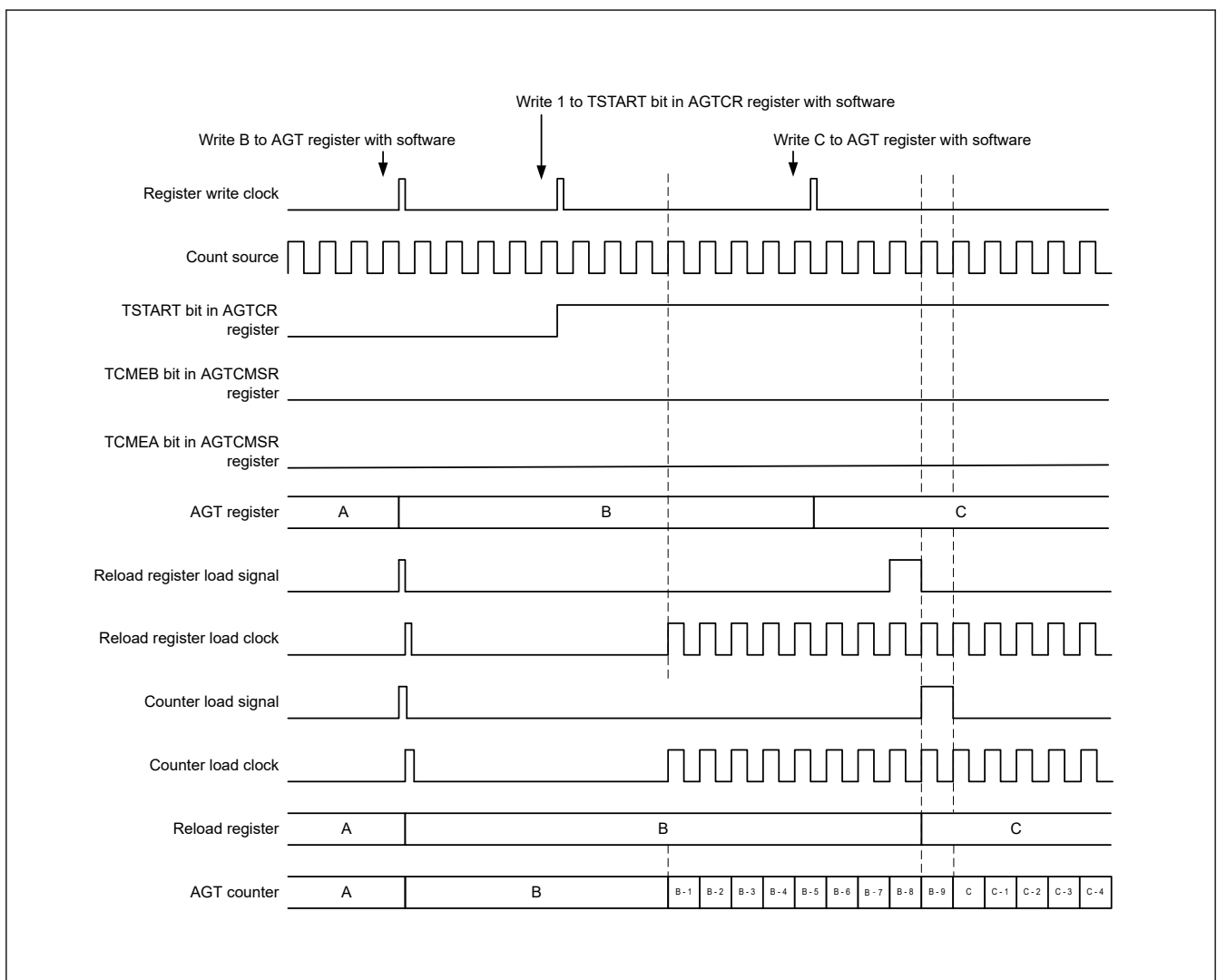
The TIES bit enables or disables an external event input.

**22.3 Operation**

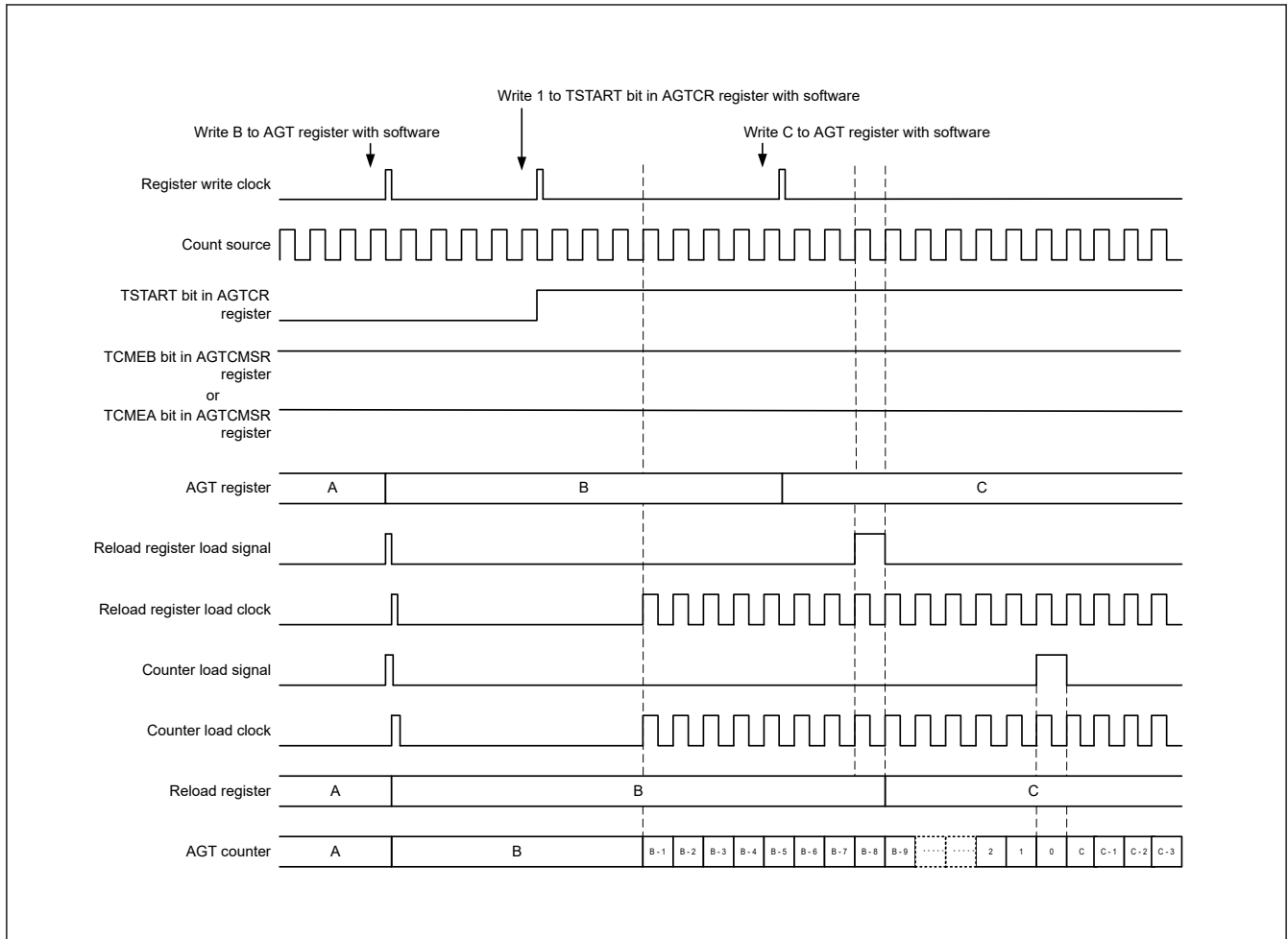
**22.3.1 Reload Register and Counter Rewrite Operation**

Regardless of the operating mode, the timing of the rewrite operation to the reload register and the counter differs depending on the value of the TSTART bit in the AGTCR register and of the TCMEA or TCMEB bit in the AGTCMSR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and the counter. When the TSTART bit is 1 (count starts) and the TCMEA bit and TCMEB bit are 0 (AGT compare match A/B register are invalid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the next count source. When the TSTART bit is 1 (count starts) and the TCMEA bit or the TCMEB bit is 1 (AGT compare match A register or compare match B register is valid), the value is written to the reload register in synchronization with the count source, and then to the counter in synchronization with the underflow of the counter.

Figure 22.3 and Figure 22.4 show the timing of rewrite operation with TSTART bit value and TCMEA/TCMEB bit value.



**Figure 22.3 Timing of rewrite operation with TSTART, TCMEA, and TCMEB bit value when AGT compare match A register and AGT compare match B register is invalid**

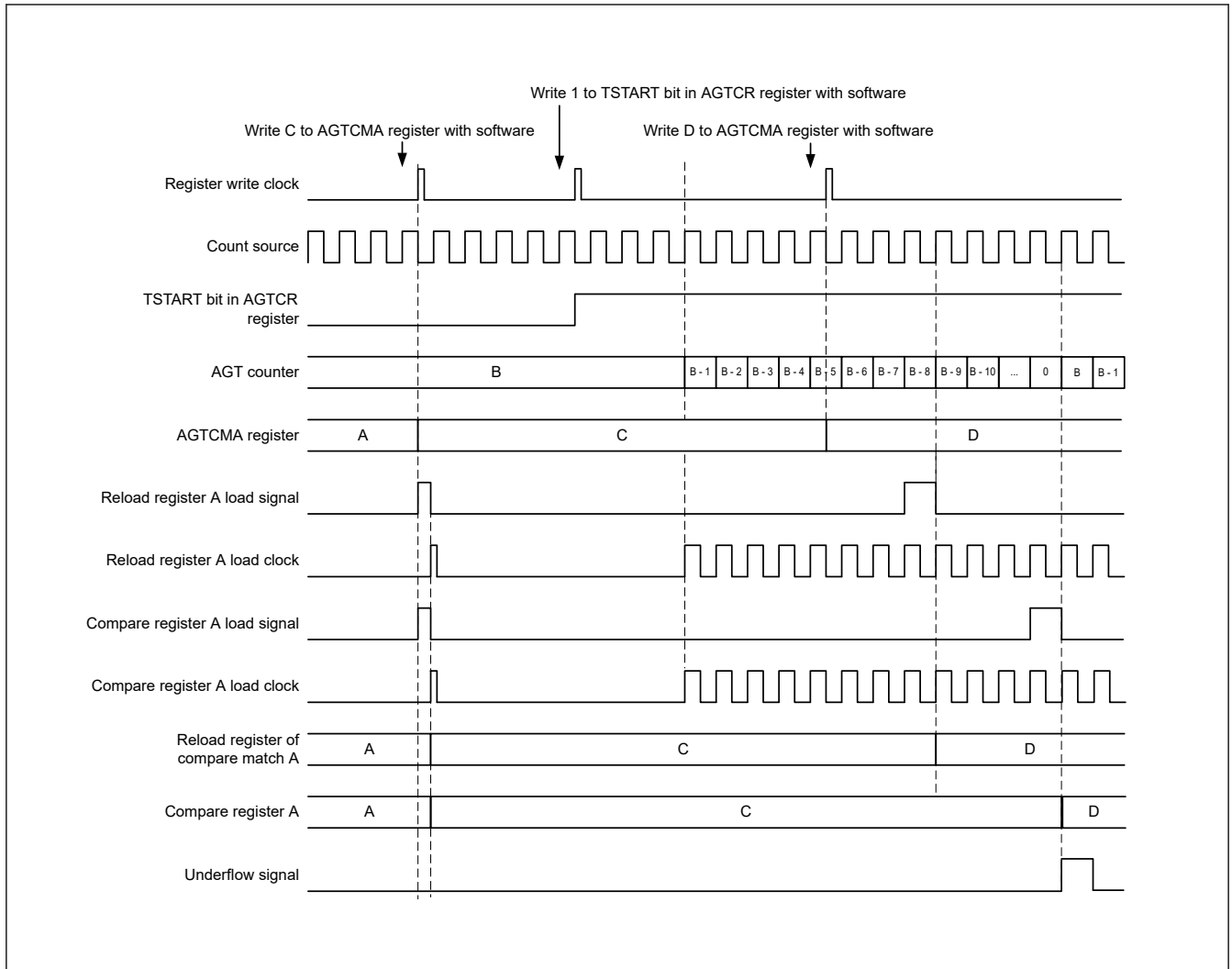


**Figure 22.4** Timing of rewrite operation with TSTART bit value and TCMEA or TCMEB bit value when AGT compare match A register or AGT compare match B register is valid

### 22.3.2 Reload Register and AGT Compare Match A/B Register Rewrite Operation

Regardless of the operating mode, the timing of the rewrite operation to the reload register and AGT compare register A/B depends on the value of the TSTART bit in the AGTCR register. When the TSTART bit is 0 (count stops), the count value is directly written to the reload register and AGT compare register A/B. When the TSTART bit is 1 (count starts), the value is written to the reload register in synchronization with the count source, and then to the compare register in synchronization with the underflow of the counter.

Figure 22.5 shows the timing of rewrite operation with TSTART bit value for compare register A. AGT Compare register B is of the same timing as AGT compare register A.

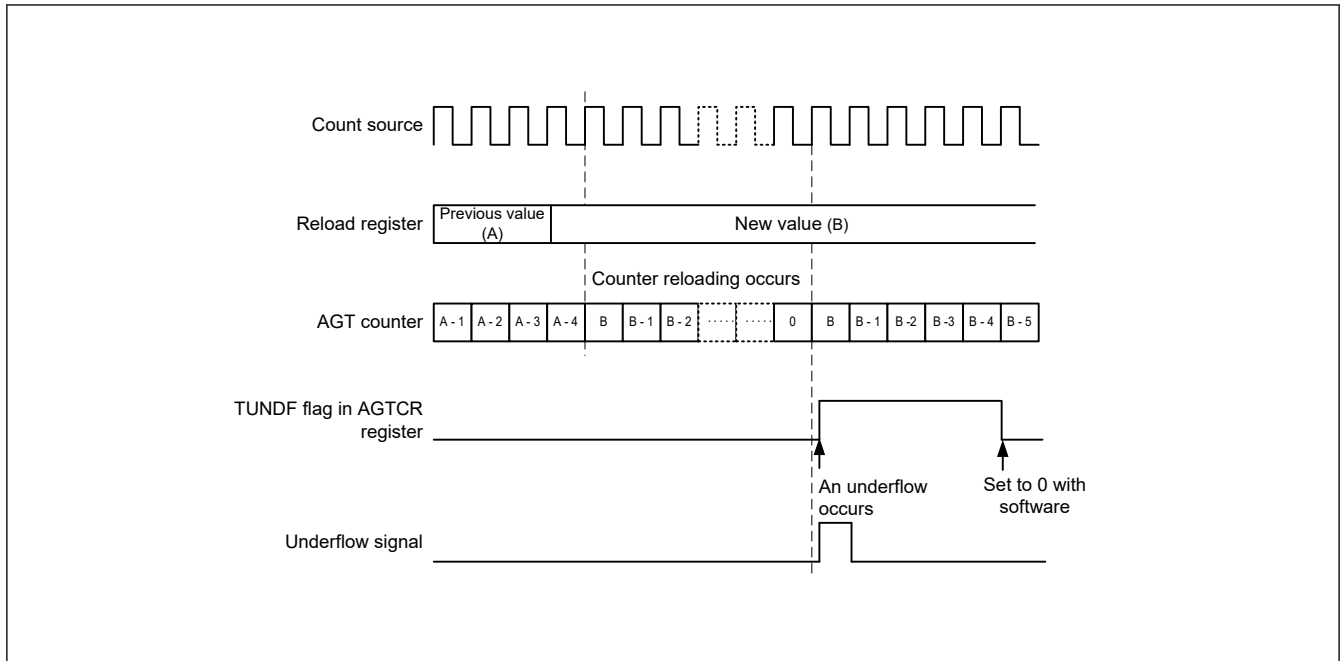


**Figure 22.5** Timing of rewrite operation with the TSTART bit value for AGT compare register A

### 22.3.3 Timer Mode

In this mode, the AGT counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register. In timer mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0x0000 and the next count source is input, an underflow occurs and an interrupt request is generated.

Figure 22.6 shows the operation example in timer mode.



**Figure 22.6** Operation example in timer mode

### 22.3.4 Pulse Output Mode

In pulse output mode, the counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and the output level of the AGTIO<sub>n</sub> and AGTON pins inverted each time an underflow occurs.

In pulse output mode, the count value is decremented by 1 on each rising edge of the count source. When the count value reaches 0x0000 and the next count source is input, an underflow occurs and an interrupt request is generated. In addition, a pulse can be output from the AGTIO<sub>n</sub> and AGTON pins. The output level is inverted each time an underflow occurs. The pulse output from the AGTON pin can be stopped with the TOE bit in the AGTIOC register. The output level can be selected with the TEDGSEL bit in the AGTIOC register.

Figure 22.7 shows the operation example in pulse output mode.

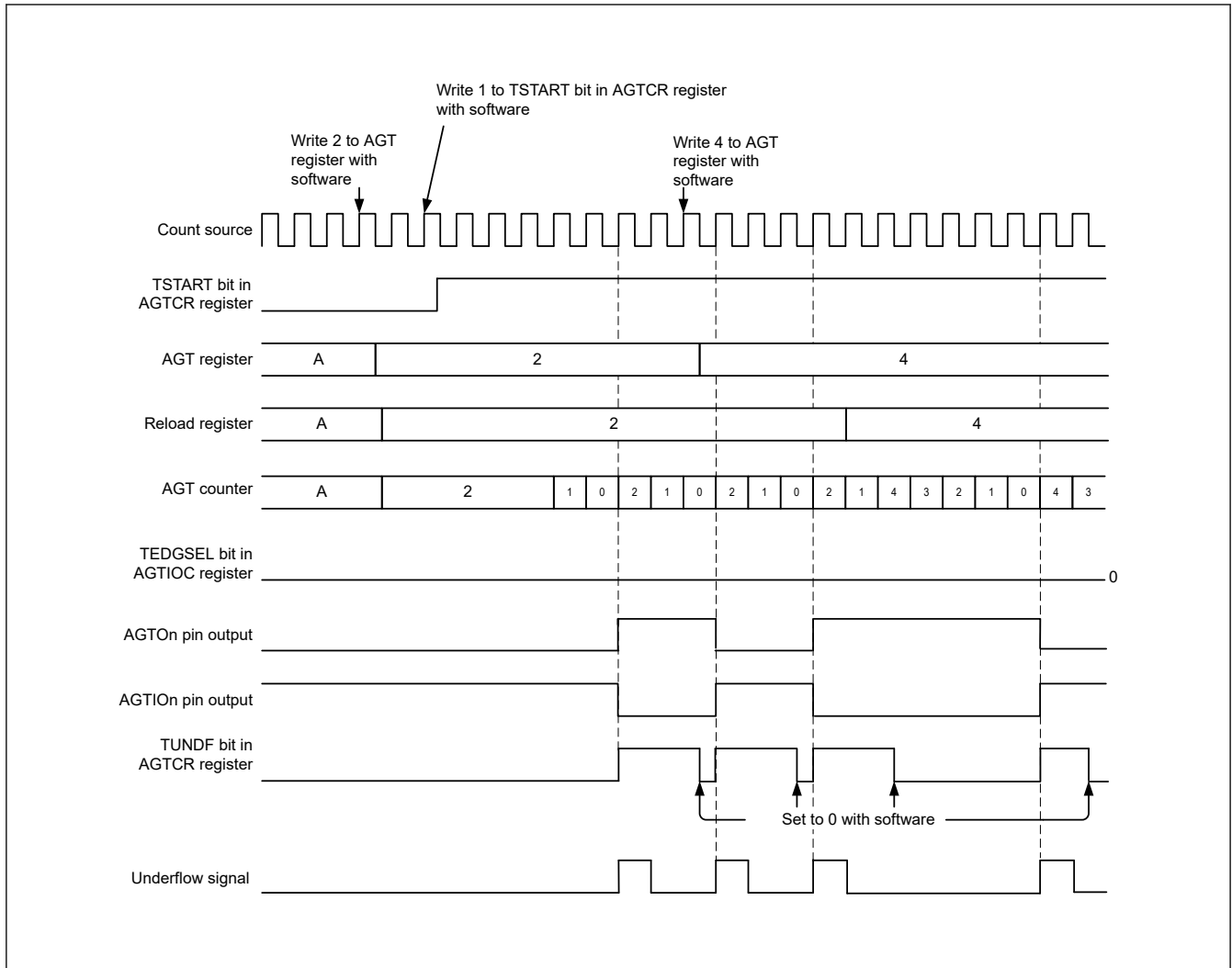
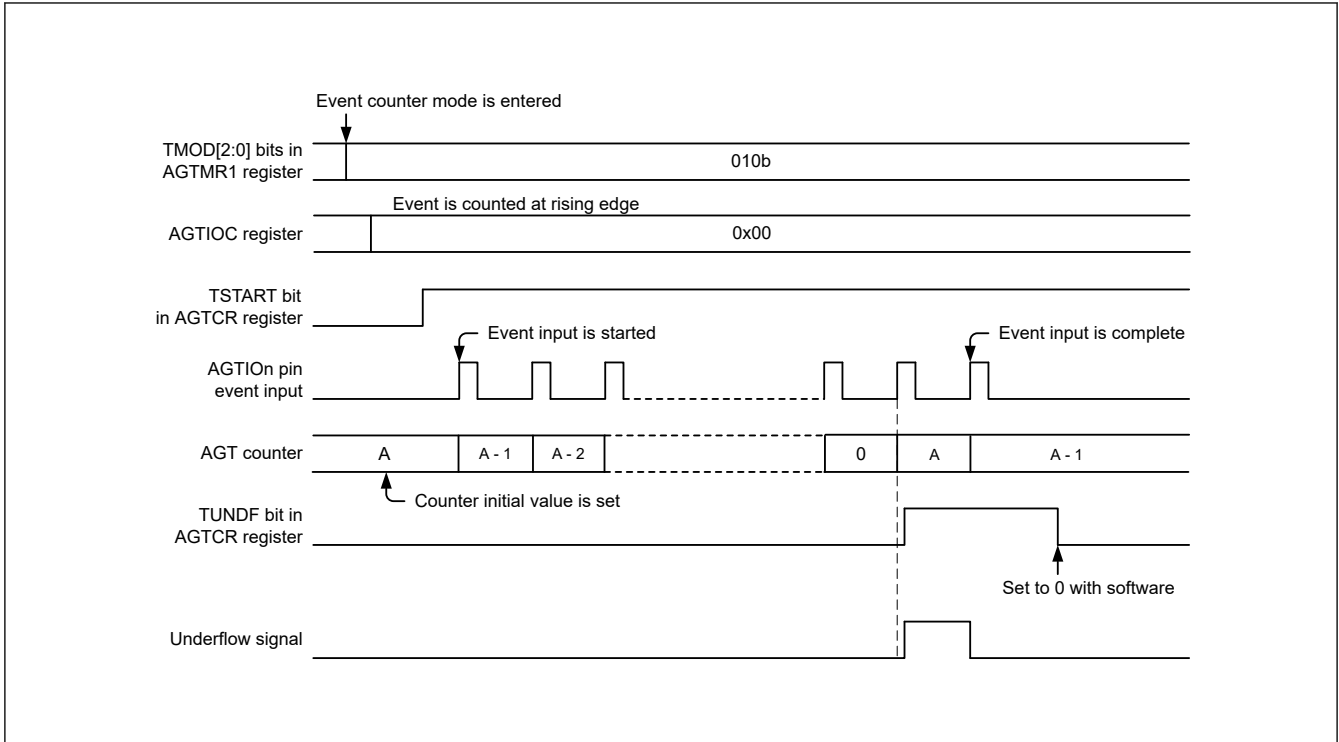


Figure 22.7 Operation example in pulse output mode

### 22.3.5 Event Counter Mode

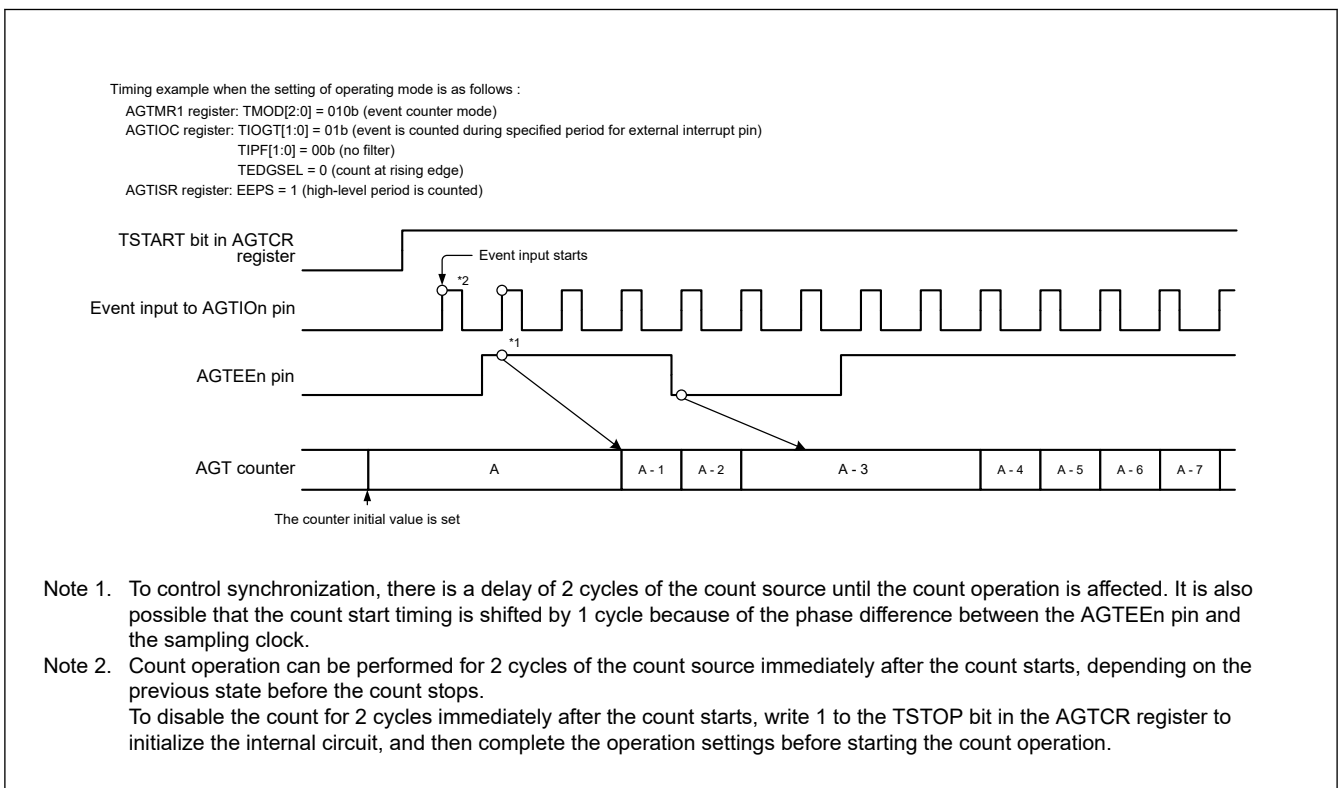
In event counter mode, the counter is decremented by an external event signal (count source) input to the AGTIO pin. Various periods for counting events can be set with the TIOGT[1:0] bits in the AGTIOC register and AGTISR registers. In addition, the filter function for the AGTIO pin input can be specified with bits TIPF[1:0] in the AGTIOC register. The output from the AGTOn pin can be toggled even in event counter mode.

Figure 22.8 shows the operation example in event counter mode.



**Figure 22.8 Operation example 1 in event counter mode**

Figure 22.9 shows an operation example for counting during the specified period in event counter mode (TIOGT[1:0] bits in the AGTIOC register are set to 01b).



**Figure 22.9 Operation example 2 in event counter mode**

### 22.3.6 Pulse Width Measurement Mode

In pulse width measurement mode, the pulse width of an external signal input to the AGTIO pin is measured. When the level specified by the TEDGSEL bit in the AGTIOC register is input to the AGTIO pin, the counter is decremented by the



count source selected with the TCK[2:0] bits in the AGTMR1 register. When the specified level on the AGTIO pin ends, the counter is stopped, the TEDGF flag in the AGTCR register is set to 1 (active edge received), and an interrupt request is generated. The measurement of pulse width data is performed by reading the count value while the counter is stopped. Also, when the counter underflows during measurement, the TUNDF flag in the AGTCR register is set to 1 and an interrupt request is generated.

Figure 22.10 shows the operation example in pulse width measurement mode.

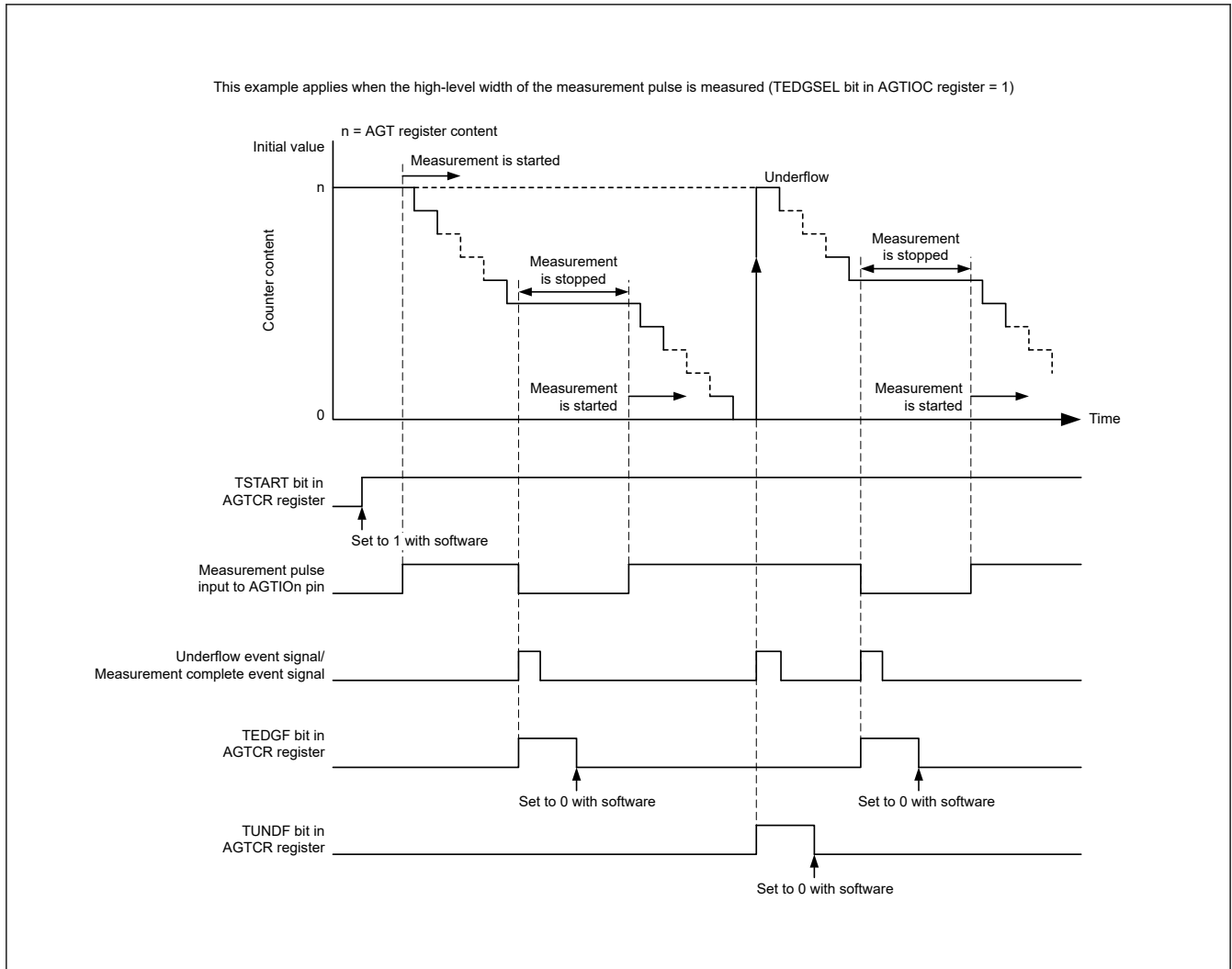


Figure 22.10 Operation example in pulse width measurement mode

### 22.3.7 Pulse Period Measurement Mode

In pulse period measurement mode, the pulse period of an external signal input to the AGTIO pin is measured. The counter is decremented by the count source selected with TCK[2:0] bits in the AGTMR1 register. When a pulse with the period specified by the TEDGSEL bit in the AGTIOC register is input to the AGTIO pin, the count value is transferred to the read-out buffer on the rising edge of the count source. The value in the reload register is loaded to the counter at the next rising edge. Simultaneously, the TEDGF flag in the AGTCR register is set to 1 (active edge received) and an interrupt request is generated. The read-out buffer (AGT register) is read at this time and the difference from the reload value (see section 22.4.6. How to Calculate Event Number, Pulse Width, and Pulse Period) is the period data of the input pulse. The period data is retained until the read-out buffer is read. When the counter underflows, the TUNDF flag in the AGTCR register is set to 1 (underflow) and an interrupt request is generated.

Figure 22.11 shows the operation example in pulse period measurement mode.

Only input pulses with a period longer than twice the period of the count source are measured. Also, the low-level and high-level widths must both be longer than the period of the count source. If a pulse period shorter than these conditions is input, the input might be ignored.

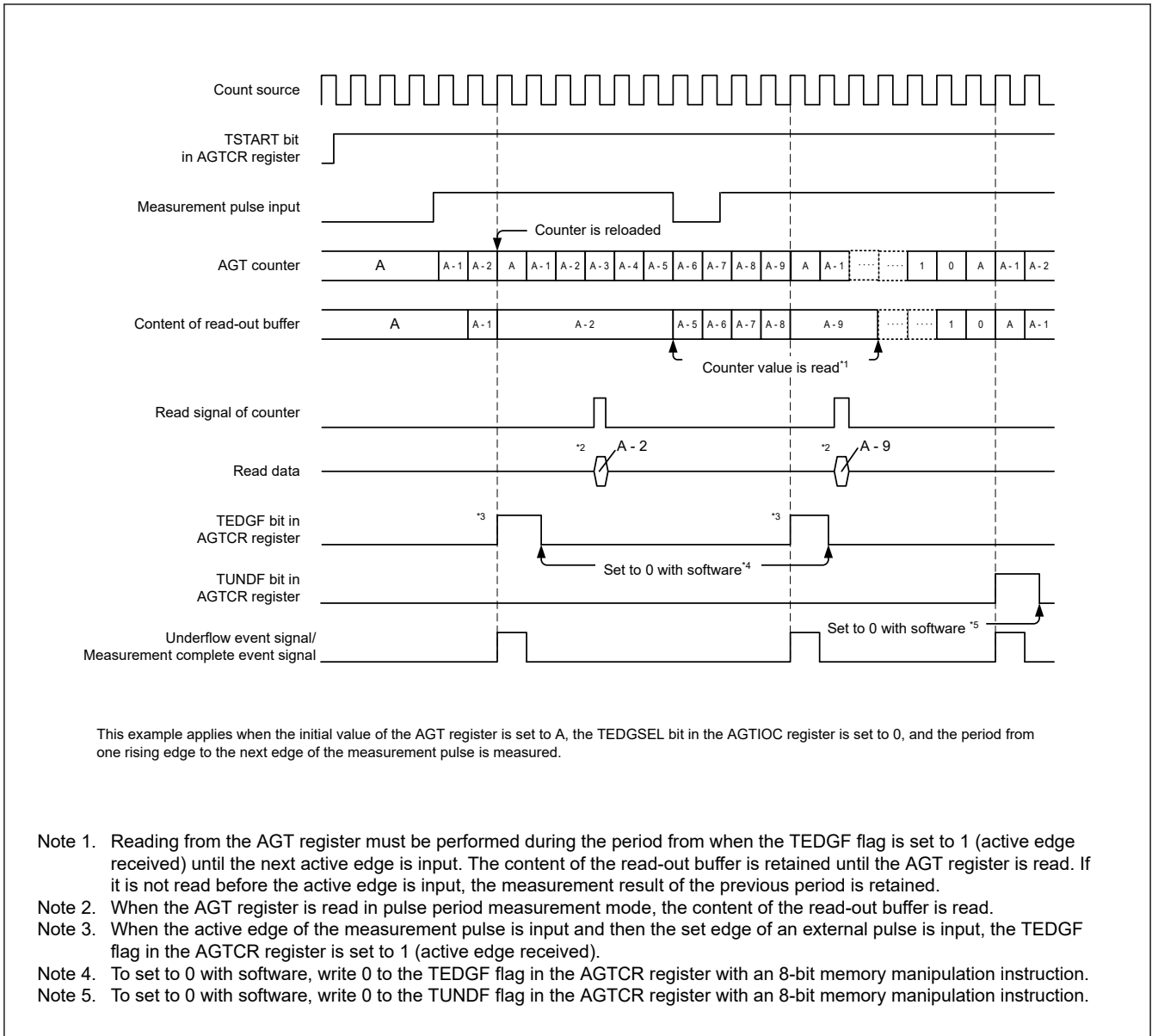


Figure 22.11 Operation example in pulse period measurement mode

### 22.3.8 Compare Match function

The compare match function detects matches (compare match) between the content of the AGTCMA or AGTCMB register and the content of the AGT register. This function is enabled when the TCMEA or TCMEB bit in the AGTCMSR register is 1 (compare match A register or compare match B register is valid). The counter is decremented by the count source selected with the TCK[2:0] bits in the AGTMR1 register, and when the values of AGT and AGTCMA or AGTCMB match, the TCMAF/TCMBF flag in the AGTCR register is set to 1 (match), and an interrupt request is generated.

When the compare match function is enabled, the timing of the rewrite operation to the reload register and the counter differs. See [section 22.3.1. Reload Register and Counter Rewrite Operation](#) for details. In addition, the output level of the AGTOAn, AGTOBn pins is inverted by the match and by the underflow. The output level can be selected with the TOPOLA or TOPOLB bit in the AGTCMSR register.

Figure 22.12 shows the operation example in compare match function.

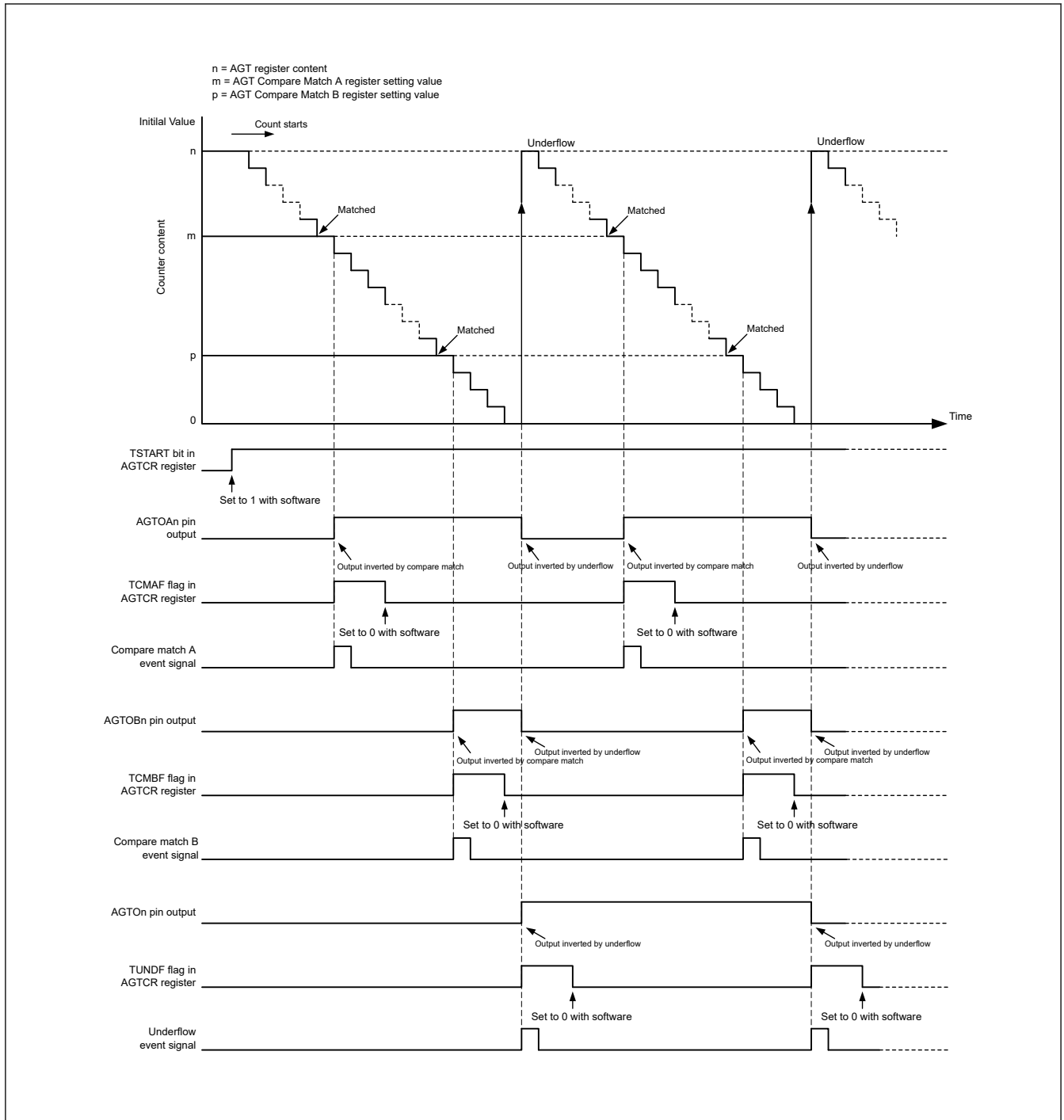


Figure 22.12 Operation example in compare match function (TOPOLA = 0, TOPOLB = 0)

### 22.3.9 Output Settings for Each Mode

Table 22.5 to Table 22.8 list the states of pins AGTON, AGTION, AGTOAn, and AGTOBn pins in each mode.

Table 22.5 AGTON pin setting

Operating mode	AGTIOC register		AGTON pin output
	TOE bit	TEDGSEL bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

**Table 22.6 AGTIO<sub>n</sub> pin setting**

Operating mode	AGTIOC register	
	TEDGSEL bit	AGTIO <sub>n</sub> pin I/O
Timer mode	0 or 1	Input (not used)
Pulse output mode	1	Normal output
	0	Inverted output
Event counter mode	0 or 1	Input
Pulse width measurement mode		
Pulse period measurement mode		

**Table 22.7 AGTOA<sub>n</sub> pin setting**

Operating mode	AGTCMSR register		AGTOA <sub>n</sub> pin output
	TOEA bit	TOPOLA bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

**Table 22.8 AGTOB<sub>n</sub> pin setting**

Operating mode	AGTCMSR register		AGTOB <sub>n</sub> pin output
	TOEB bit	TOPOLB bit	
Timer mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse output mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Event counter mode	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled (not used)
Pulse width measurement mode	0	0	Prohibited
Pulse period measurement mode			

### 22.3.10 Standby Mode

The AGT can operate in Software Standby mode. Set it to Software Standby mode with count operation start (TSTART = 1, TCSTF = 1).

[Table 22.9](#) and [Table 22.10](#) show the setting that can be used in Software Standby mode.

**Table 22.9 Usable settings in Software Standby mode (AGT0)**

Operating mode	AGTMR1.TCK[2:0]	Operating clock	Resurgence factor of CPU
Timer mode	100b or 110b	AGTLCLK or AGTSCLK	—
Pulse output mode	100b or 110b	AGTLCLK or AGTSCLK	—
Event counter mode <sup>*2</sup>	—	AGTIO0 <sup>*1</sup>	—
Pulse width measurement mode	100b or 110b	AGTLCLK or AGTSCLK	—
Pulse period measurement mode	100b or 110b	AGTLCLK or AGTSCLK	—

Note: —: invalid

Note 1. When using the AGTIO0 pin for external event input in Software Standby mode, set AGTIOSEL.TIES = 1.

Note 2. AGTEE pin is not available during Software Standby mode. External events are always enabled.

**Table 22.10 Usable settings in Software Standby mode (AGT1)**

Operating mode	AGTMR1.TCK[2:0]	Operating clock	Resurgence factor of CPU
Timer mode	100b or 110b or 101b <sup>*1</sup>	AGTLCLK or AGTSCLK or AGT0 underflow	<ul style="list-style-type: none"> <li>Underflow</li> <li>Compare match A/B</li> </ul>
Pulse output mode	100b or 110b or 101b <sup>*1</sup>	AGTLCLK or AGTSCLK or AGT0 underflow	<ul style="list-style-type: none"> <li>Underflow</li> <li>Compare match A/B</li> </ul>
Event counter mode <sup>*3</sup>	—	AGTIO1 <sup>*2</sup>	<ul style="list-style-type: none"> <li>Underflow</li> <li>Compare match A/B</li> </ul>
Pulse width measurement mode	100b or 110b or 101b <sup>*1</sup>	AGTLCLK or AGTSCLK or AGT0 underflow	<ul style="list-style-type: none"> <li>Underflow</li> <li>Active edge</li> </ul>
Pulse period measurement mode	100b or 110b or 101b <sup>*1</sup>	AGTLCLK or AGTSCLK or AGT0 underflow	<ul style="list-style-type: none"> <li>Underflow</li> <li>Active edge</li> </ul>

Note: —: invalid

Note: Release of Software Standby mode is only AGT1.

Note: Compare match A/B is resurgence factor of CPU from Software Standby mode.

Note 1. Only when AGT0 operates in [Table 22.9](#)

Note 2. When using the AGTIO1 pin for external event input in Software Standby mode, set AGTIOSEL.TIES = 1.

Note 3. AGTEE pin is not available during Software Standby mode. External events are always enabled

### 22.3.11 Interrupt Sources

The AGTn has three interrupt sources as listed in [Table 22.11](#).

**Table 22.11 AGT interrupt sources**

Name	Interrupt source	DMAC/DTC activation
AGTn_AGTI	<ul style="list-style-type: none"> <li>When the counter underflows</li> <li>When measurement of the active width of the external input pin (AGTIO<sub>n</sub>) is complete in pulse width measurement mode</li> <li>When the set edge of the external input pin (AGTIO<sub>n</sub>) is input in pulse period measurement mode.</li> </ul>	Possible
AGTn_AGTCMAI	<ul style="list-style-type: none"> <li>When the values of AGT register and AGTCMA register match</li> </ul>	Possible
AGTn_AGTCMBI	<ul style="list-style-type: none"> <li>When the values of AGT register and AGTCMB register match</li> </ul>	Possible

Note: Channel number (n = 0, 1)

### 22.3.12 Event Signal Output to ELC

The AGT uses the Event Link Controller (ELC) to perform a link operation to a specified module using the interrupt request signal as the event signal. The AGT outputs compare match A, compare match B, and underflow/measurement complete signals as event signals. For details, see [section 18, Event Link Controller \(ELC\)](#).

## 22.4 Usage Notes

### 22.4.1 Count Operation Start and Stop Control

- When the operating mode (see [Table 22.1](#)) is set to other than the event counter mode, or the count source is set to other than AGTn underflow event signal (TCK[2:0] = 101b):
  - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 3 cycles of the count source. Do not access the registers associated with AGT other than the TCSTF flag until this flag is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF flag remains 1 for 3 cycles of the count source. When the TCSTF flag is set to 0, the count is stopped. Do not access the registers associated with AGT other than the TCSTF flag until this flag is set to 0.
- When the operating mode (see [Table 22.1](#)) is set to event counter mode, or the count source is set to AGT1 underflow event signal (TCK[2:0] = 101b):
  - After 1 (count starts) is written to the TSTART bit in the AGTCR register while the count is stopped, the TCSTF flag in the AGTCR register remains 0 (count stops) for 2 PCLKB cycles. Do not access the registers associated with AGT other than the TCSTF flag until this flag is set to 1 (count in progress).
  - After 0 (count stops) is written to the TSTART bit during a count operation, the TCSTF flag remains 1 for 2 PCLKB cycles. When the TCSTF flag is set to 0, the count is stopped. Do not access the registers associated with AGT other than the TCSTF flag until this flag is set to 0.

### 22.4.2 Access to Counter Register

When the TSTART bit and TCSTF flag in the AGTCR register are both 1 (count starts), allow at least 3 cycles of the count source clock between writes when writing to the AGT register successively.

### 22.4.3 When Changing Mode

The registers associated with AGT operating mode (AGTMR1, AGTMR2, AGTIOC, AGTISR, and AGTCMSR) can be changed only when the count is stopped with both the TSTART bit and TCSTF flag set to 0 (count stops). Do not change these registers during count operation.

When the registers associated with AGT operating mode are changed, the values of TEDGF, TUNDF, TCMAF, and TCMBF flags are undefined. Before starting the count, write 0 to the following flags:

- TEDGF (no active edge received)
- TUNDF (no underflow)
- TCMAF (no match)
- TCMBF (no match).

### 22.4.4 Output pin setting

When using the AGTOn, AGTIO<sub>n</sub>, AGTOAn, or AGTOB<sub>n</sub> as an output pin, set up the Operation and determine the initial output values. Then set PmnPFS.PMR bit to 1.

When using the AGTIO<sub>n</sub> as an input pin in pulse width measurement mode or pulse period measurement mode, set up the Operation and start count operation. Then start to enter external events from the AGTIO<sub>n</sub> pin. Invalidate the first measurement and validate the second and later completed measurements.

### 22.4.5 Digital Filter

When using the digital filter, do not start the timer operation for 5 cycles of the digital filter clock after setting TIPF[1:0] bits and when the TEDGSEL bit in the AGTIOC register changes.

### 22.4.6 How to Calculate Event Number, Pulse Width, and Pulse Period

- In event counter mode, event number is expressed mathematically as follows:  
Event number = initial value of counter [AGT register] - counter value of active event end

- In pulse width measurement mode, pulse width is expressed mathematically as follows:  
Pulse width = counter value of stopping measurement - counter value of next stopping measurement
- In pulse period measurement mode, input pulse period is expressed mathematically as follows:  
Period of input pulse = (initial value of counter [AGT register] - reading value of the read-out buffer) + 1.

#### 22.4.7 When Count is Forcibly Stopped by TSTOP Bit

After the counter is forcibly stopped by the TSTOP bit in the AGTCR register, do not access the following I/O registers for 1 cycle of the count source:

- AGT
- AGTCMA
- AGTCMB
- AGTCR
- AGTMR1
- AGTMR2.

#### 22.4.8 When Selecting AGT0 Underflow as the Count Source

Operate according to the following procedures described in this section when selecting the underflow event signal as the count source.

##### (1) Procedure for starting operation

1. Set AGT.
2. Start the count operation of AGT1.
3. Start the count operation of AGT0.

##### (2) Procedure for stopping operation

1. Stop the count operation of AGT0.
2. Stop the count operation of AGT1.
3. Stop the count source clock of AGT1 (write 000b in the AGTMR1.TCK[2:0] bits).

#### 22.4.9 Module-stop function

AGT operation can be disabled or enabled using Module Stop Control Register D (MSTPCRD). The AGT module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#)

#### 22.4.10 When Switching Source Clock

When switching a clock source by changing SCKSCR.CKSEL[2:0], the clock output from the selector stops for 4 cycles of the switched clock. Therefore, when using the AGTIO<sub>n</sub>, AGTEEn, or both input as external event input, the clock source should not be switched. If switching the clock source while using the external event input, extend the input pulse width by 4 clock cycles of the switched source clock cycles.

## 23. Ultra-Low-Power Timer (ULPT)

### 23.1 Overview

This ultra-low-power timer (ULPT) is a 32-bit timer that can output pulses or count external events.

This 32-bit timer consists of reload registers and a down-counter. The reload registers and the down-counter are allocated to the same address and can be accessed through the ULPTCNT register.

Table 23.1 lists the ULPT specifications.

**Table 23.1 ULPT specifications**

Item		Description
Operating mode	Timer mode	Count the internal clock.
	Event counter mode	Count external events.
Number of Channels		32 bits × 2 channels (ULPTn ( n = 0, 1 ))
Count source (operating clock)*1		The following signals can be selected: <ul style="list-style-type: none"> <li>Internal clock: ULPTLCLK/d or ULPTSCLK/d (d = 1, 2, 4, 8, 16, 32, 64, 128)</li> <li>External event: ULPTEVIn pin</li> </ul>
Count operation		The following functions can be selected: <ul style="list-style-type: none"> <li>Selection between continuous mode and one-shot mode</li> <li>Selection among count enable mode, count start mode, and count restart mode</li> </ul>
Interrupt/event link function (output)		<ul style="list-style-type: none"> <li>Underflow signal <ul style="list-style-type: none"> <li>When the counter underflows</li> </ul> </li> <li>Compare match A signal <ul style="list-style-type: none"> <li>When the ULPTCNT value matches the ULPTCMA value (The compare match A function is enabled.)</li> </ul> </li> <li>Compare match B signal <ul style="list-style-type: none"> <li>When the ULPTCNT value matches the ULPTCMB value (The compare match B function is enabled.)</li> </ul> </li> </ul>
Selectable functions		<ul style="list-style-type: none"> <li>Pulse output function <ul style="list-style-type: none"> <li>Inverts the output each time the counter underflows.</li> </ul> </li> <li>Compare match function <ul style="list-style-type: none"> <li>Either or both compare match A register and compare match B register are selectable.</li> </ul> </li> <li>Return from Software Standby mode can be performed with ULPTn_ULPTI, ULPTn_ULPTCMAI, or ULPTn_ULPTCMBI (n=0, 1).*2</li> <li>Return from Deep Software Standby mode 1 can be performed with ULPTn_ULPTI (n=0, 1).*2</li> </ul>
Module-stop function		Module-stop state can be set for each channels to reduce power consumption
TrustZone Filter		Security and Privilege attribution can be set for each channels.

Note 1. Make sure that the frequency of the peripheral module clock (PCLKB) ≥ the frequency of the count source clock.

Note 2. For details, see [section 10, Low Power Modes](#).

### 23.2 Configuration

Figure 23.1 shows the ULPT block diagram. Table 23.2 lists the I/O pins of the ULPT.



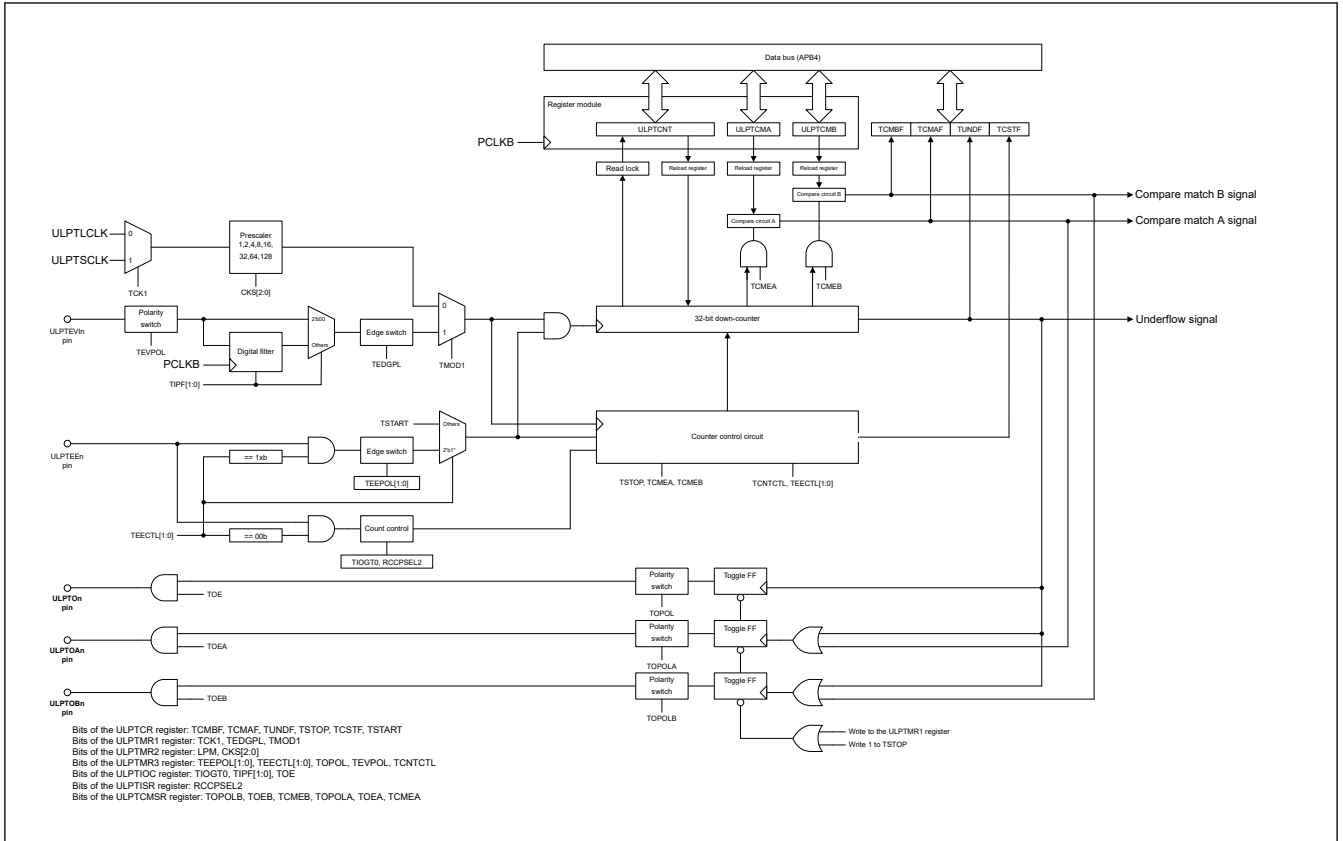


Figure 23.1 ULPT block diagram

Table 23.2 ULPT I/O pins

Pin name	I/O	Function
ULPTEEn <sup>*2</sup>	Input	External count control input <sup>*1</sup>
ULPTEVIn <sup>*2</sup>	Input	External event input
ULPTOn <sup>*2</sup>	Output	Pulse output
ULPTOAn <sup>*2</sup>	Output	Output compare match A output
ULPTOBn <sup>*2</sup>	Output	Output compare match B output

Note: Channel number: n = 0, 1

Note 1. The function of the ULPTEEn pin can be selected by the ULPTMR3.TEECTL[1:0] bits. For more details, see [section 23.3.7. ULPTMR3 : ULPT Mode Register 3.](#)

Note 2. Only pins marked "-DS" can be used in Deep Software Standby mode1.

### 23.3 Register Descriptions

#### 23.3.1 ULPTCNT : ULPT Counter Register

Base address: ULPTn = 0x4022\_0000 + 0x0100 × n (n = 0, 1)  
 ULPTn\_NS = 0x5022\_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position: 31

Bit field:



Value after reset: 1

Bit	Symbol	Function	R/W
31:0	ULPTCNT	32-bit counter and reload register <sup>*1*2</sup> Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. When 1 is written to the ULPTCR.TSTOP bit, the counter is forcibly stopped and set to 0xFFFFFFFF. The reload register is set to 0xFFFFFFFF

Note 2. When 0x00000000 is set in the ULPTCNT register, a request signal to the ICU, DTC, and ELC occurs only once immediately after the start of count operation (when the ULPTCR.TSTART bit is set to 1). However, the pulse output at the ULPTOn pin continues to be toggled.

The pulse output from the ULPTOn pin can be enabled or disabled by the ULPTIOC.TOE bit.

When the ULPTCNT register is set to a value equal to or greater than 0x00000001, a request signal is generated each time the counter underflows.

The ULPTCNT is a 32-bit register. The register value is written to the reload register and is read from the counter.

The states of the counter and reload register change according to the TSTART/TCSTF bit in the ULPTCR register and TCMEA/TCMEB bit in the ULPTCMSR register. For details, see [section 23.4.2. Rewriting the counter and reload register](#).

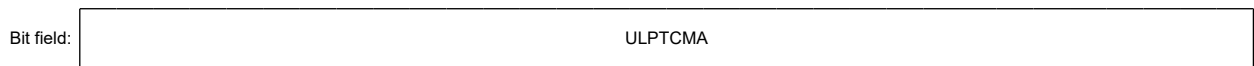
The ULPTCNT register is only available for 32-bit read/write access.

### 23.3.2 ULPTCMA : ULPT Compare Match A Register

Base address:  $ULPTn = 0x4022\_0000 + 0x0100 \times n$  (n = 0, 1)  
 $ULPTn\_NS = 0x5022\_0000 + 0x0100 \times n$  (n = 0, 1)

Offset address: 0x04

Bit position: 31 0



Value after reset: 1

Bit	Symbol	Function	R/W
31:0	ULPTCMA	32-bit Compare Match A Data <sup>*1*2</sup> Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. When 1 is written to the ULPTCR.TSTOP bit, 0xFFFFFFFF is set in the ULPTCMA register.

Note 2. Set the ULPTCMA register to 0xFFFFFFFF when compare match A is not used (ULPTCMSR.TCMEA bit = 0).

The ULPTCMA register is a readable/writable register to set a value for compare match with the counter.

The states of compare circuit A and the reload register change according to the TSTART and TCSTF bits in the ULPTCR register. For details, see [section 23.4.3. Rewriting the compare circuits and reload registers for compare match A/B](#).

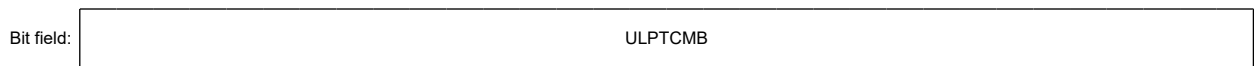
The ULPTCMA register is only available for 32-bit read/write access.

### 23.3.3 ULPTCMB : ULPT Compare Match B Register

Base address:  $ULPTn = 0x4022\_0000 + 0x0100 \times n$  (n = 0, 1)  
 $ULPTn\_NS = 0x5022\_0000 + 0x0100 \times n$  (n = 0, 1)

Offset address: 0x08

Bit position: 31 0



Value after reset: 1

Bit	Symbol	Function	R/W
31:0	ULPTCMB	32-bit Compare Match B Data <sup>*1*2</sup> Setting range : 0x00000000 to 0xFFFFFFFF	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. When 1 is written to the ULPTCR.TSTOP bit, 0xFFFFFFFF is set in the ULPTCMB register.

Note 2. Set the ULPTCMB register to 0xFFFFFFFF when compare match B is not used (ULPTCMSR.TCMEB bit = 0).

The ULPTCMB register is a readable/writable register to set a value for compare match with the counter.

The states of compare circuit B and the reload register change according to the TSTART and TCSTF bits in the ULPTCR register. For details, see [section 23.4.3. Rewriting the compare circuits and reload registers for compare match A/B](#).

The ULPTCMB register is only available for 32-bit read/write access.

### 23.3.4 ULPTCR : ULPT Control Register

Base address:  $ULPTn = 0x4022\_0000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $ULPTn\_NS = 0x5022\_0000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCMB F	TCMA F	TUNDF F	—	—	TSTOP P	TCSTF F	TSTART RT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSTART	Counter Start 0: Stop the counter. 1: Start the counter.	R/W
1	TCSTF	Counter Status Flag* <sup>2</sup> 0: Counter stopped 1: Counter running	R
2	TSTOP	Counter Forcible Stop* <sup>1</sup> 0: Writing is invalid. 1: Stop the counter forcibly.	W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	TUNDF	Underflow Flag 0: No underflow occurred (counter $\neq$ 0x00000000). 1: Underflow occurred (counter = 0x00000000).	R/W* <sup>3</sup>
6	TCMAF	Compare Match A Flag 0: Not matched (counter $\neq$ ULPTCMA[31:0]) 1: Matched (counter = ULPTCMA[31:0])	R/W* <sup>3</sup>
7	TCMBF	Compare Match B Flag 0: Not matched (counter $\neq$ ULPTCMB[31:0]) 1: Matched (counter = ULPTCMB[31:0])	R/W* <sup>3</sup>

Note: S-TYPE-3, P-TYPE-3

Note 1. When 1 (stop the counter forcibly) is written to the TSTOP bit, the TSTOP, TSTART, and TCSTF bits are initialized. The pulse output, compare match A output and compare match B output are also initialized. This bit is read as 0.

Note 2. For precautions on using the TCSTF bit, see [section 23.5.1. Start and stop control of the counter](#).

Note 3. Only 0 can be written to clear the flag.

The ULPTCR register controls the start and stop of the counter and indicates the status of the ULPT.

The ULPTCR register is only available for 8-bit read/write access.

It is recommended to set 1 to the TUNDF, TCMAF and TCMBF bits when changing the TSTART or TSTOP bits to prevent clearing TUNDF, TCMAF, or TCMBF by mistake. At this time, setting 1 to TUNDF, TCMAF and TCMBF bits is ignored.

#### TSTART bit (Counter Start)

When and how the counter is started depends on the setting of the ULPTMR3.TEECTL[1:0] bits.

The counter is stopped in the same way in any mode.

For details, see [section 23.4.1. Count operation](#).

[Starting the counter]

- In count enable mode (ULPTMR3.TEECTL[1:0] bits = 00)  
When 1 is written to the TSTART bit.
- In count start mode or count restart mode (ULPTMR3.TEECTL[1:0] bits = 10 or 11)

When 1 is written to the TSTART bit, and then an edge at the ULPTEEn pin is input.

[Stopping the counter]

- When 0 is written to the TSTART bit.
- When 1 is written to the TSTOP bit.

### TCSTF bit (Counter Status Flag)

The operation of this bit depends on the settings of the TCNTCTL and TEECTL[1:0] bits of the ULPTMR3 register.

For details, see [section 23.4.1. Count operation](#).

[Setting conditions]

- In timer mode (ULPTMR1.TMOD1 = 0) and count enable mode (ULPTMR3.TEECTL[1:0] bits = 00)  
When 1 is written to the TSTART bit (with the TCSTF bit set to 1 in synchronization with the count source).
- In event counter mode (ULPTMR1.TMOD1 = 1) and count enable mode (ULPTMR3.TEECTL[1:0] bits = 00)  
When 1 is written to the TSTART bit (with the TCSTF bit set to 1 in synchronization with the count source).
- In count start mode or count restart mode (ULPTMR3.TEECTL[1:0] bits = 10 or 11)  
When 1 is written to the TSTART bit, and then an edge at the ULPTEEn pin is input (with the TCSTF bit set to 1 in synchronization with the count source).

[Clearing conditions]

- When 0 is written to the TSTART bit (with the TCSTF bit set to 0 in synchronization with the count source).
- When 1 is written to the TSTOP bit.
- When the counter underflows in one-shot mode (ULPTMR3.TCNTCTL = 1).  
Only under this condition, the TSTART bit does not change from 1.

### TSTOP bit (Counter Forcible Stop)

When 1 is written to this bit, the counter is forcibly stopped. This bit is read as 0.

### TUNDF bit (Underflow Flag)

[Setting condition]

- When the counter underflows (counter = 0x00000000).

[Clearing condition]

- When 0 is written to this bit.

### TCMAF bit (Compare Match A Flag)

[Setting condition]

- When the ULPCNT register value matches the ULPCMA register value (counter = ULPCMA[31:0]).

[Clearing condition]

- When 0 is written to this bit.

### TCMBF bit (Compare Match B Flag)

[Setting condition]

- When the ULPCNT register value matches the ULPCMB register value (counter = ULPCMB[31:0]).

[Clearing condition]

- When 0 is written to this bit.

### 23.3.5 ULPTMR1 : ULPT Mode Register 1

Base address:  $ULPTn = 0x4022\_0000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $ULPTn\_NS = 0x5022\_0000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	TCK1	—	TEDGPL	—	TMOD1	—

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	TMOD1	Operating Mode*1 0: Timer mode 1: Event counter mode	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	TEDGPL	ULPTEVIn Edge Polarity 0: Either edge (rising) 1: Both edges This bit is valid only in event counter mode (TMOD1 = 1).	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	TCK1	Count Source*2 0: Divided clock specified by the ULPTMR2.CKS[2:0] bits (ULPTLCLK) 1: Divided clock specified by the ULPTMR2.CKS[2:0] bits (ULPTSCLK) This bit is valid only in timer mode (TMOD1 = 0).	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).

Note: A write access to this register initializes the output from pins ULPTOn, ULPTOAn, and ULPTOBn. For details on the output level at initialization, see [section 23.3.7. ULPTMR3 : ULPT Mode Register 3](#) and [section 23.3.8. ULPTIOC : ULPT I/O Control Register](#).

Note 1. Timer mode: The internal clock (ULPTLCLK or ULPTSCLK) is used as the count source. The counter is decremented by 1 on each rising edge of the count source.

Event counter mode: The external event input (ULPTEVIn pin) is used as the count source. The counter is decremented on each edge of the count source. The edge polarity can be selected by the TEDGPL bit and the ULPTMR3.TEVPOL bit.

For details, see [section 23.4.1. Count operation](#).

Note 2. The TCK1 bit can be changed only when the ULPTMR2.CKS[2:0] bits are 000. Otherwise, do not change the bit. Set the ULPTMR2.CKS[2:0] bits to 000, and then change the TCK1 bit after one cycle of the count source.

The ULPTMR1 register sets the operating mode of the ULPT.

The ULPTMR1 register is only available for 8-bit read/write access.

### 23.3.6 ULPTMR2 : ULPT Mode Register 2

Base address:  $ULPTn = 0x4022\_0000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $ULPTn\_NS = 0x5022\_0000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x0E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LPM	—	—	—	—	CKS[2:0]		

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	CKS[2:0]	ULPTLCLK/ULPTSCLK Count Source Clock Division Ratio <sup>*1*2</sup> 0 0 0: 1/1 0 0 1: 1/2 0 1 0: 1/4 0 1 1: 1/8 1 0 0: 1/16 1 0 1: 1/32 1 1 0: 1/64 1 1 1: 1/128 This bit is valid only in timer mode (ULPTMR1.TMOD1 = 0)	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	LPM	Low Power Mode 0: Normal mode 1: Low power mode	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).

Note 1. The CKS[2:0] bits can be changed only in timer mode (ULPTMR1.TMOD1 = 0).  
 Set the CKS[2:0] bits to 000 in event counter mode (ULPTMR1.TMOD1 = 1).

Note 2. The ULPTMR1.TCK1 bit can be changed only when the CKS[2:0] bits are 000. Set the CKS[2:0] bits to 000, and then change the ULPTMR1.TCK1 bit after one cycle of the count source.

The ULPTMR2 register sets the operating mode of the ULPT.

The ULPTMR2 register is only available for 8-bit read/write access.

### LPM bit (Low Power Mode)

This bit implements low power consumption by restricting the register access. Set this bit to 1 for low-power mode operation.

When this bit is 1, access to the following registers is prohibited:

- ULPTCNT, ULPTCMA, ULPTCMB, and ULPTCR

After this bit is switched from 1 to 0, the first access to an access-prohibited register is restricted as follows:

- ULPTCNT: The register must be read twice. Only the second reading of data is valid.
- ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR: In write access, leave two cycles of the count source.

### 23.3.7 ULPTMR3 : ULPT Mode Register 3

Base address: ULPTn = 0x4022\_0000 + 0x0100 × n (n = 0, 1)  
 ULPTn\_NS = 0x5022\_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TEEPOL[1:0]	TEECTL[1:0]	—	TOPO L	TEVP OL	TCNT CTL		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCNTCTL	Count Function Select <sup>*1*2</sup> 0: Continuous mode 1: One-shot mode	R/W
1	TEVPOL	ULPTEVIn Polarity Switch 0: External event input (ULPTEVIn pin) 1: External event input (ULPTEVIn pin) in reverse This bit is valid only in event counter mode (ULPTMR1.TMOD1 = 1).	R/W

Bit	Symbol	Function	R/W
2	TOPOL	ULPTOn Polarity Select* <sup>3</sup> 0: Start the ULPTOn output with low level. 1: Start the ULPTOn output with high level.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TEECTL[1:0]	ULPTEEn Function Select* <sup>1</sup> 0 0: Count enable mode 1 0: Count start mode 1 1: Count restart mode Other settings are prohibited. The count enable mode is valid only in event counter mode (ULPTMR1.TMOD1 = 1).	R/W
7:6	TEEPOL[1:0]	ULPTEEn Edge Polarity Select 0 0: Rising edge 0 1: Falling edge 1 0: Both edges Other settings are prohibited. This bit is valid only in count start mode or count restart mode. (TEECTL[1:0] = 10 or 11)	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).

Note 1. For details, see [section 23.4.1. Count operation](#).

Note 2. Continuous mode: After starting counting, the counter continues to count down (reload value to underflow) repeatedly until the counter stops.

One-shot mode: After starting counting, the counter counts down once (reload value to underflow).

Note 3. When data is set to the ULPTMR1 register or 1 is set to the ULPTCR.TSTOP bit, the ULPTOn output is initialized.

The ULPTMR3 register sets the operating mode of the ULPT.

The ULPTMR3 register is only available for 8-bit read/write access.

### TEECTL[1:0] bits (ULPTEEn Function Select)

The ULPTEEn pin functions differently depending on the mode. For details, see [section 23.4.1. Count operation](#).

In count enable mode, external events are counted while the ULPTEEn pin is valid. (The duration can be selected by the ULPTIOC.TIOGT0 bit, and the polarity can be selected by the ULPTISR.RCCPSEL2 bit.)

In count start mode, the counter starts counting at an edge trigger at the ULPTEEn pin. The operation of the ULPTEEn pin differs depending on the TSTART and TCSTF bits of the ULPTCR register.

In count restart mode, the counter starts counting at the first edge trigger at the ULPTEEn pin and resets the count at the second or following edge trigger. The operation of the ULPTEEn pin differs depending on the TSTART and TCSTF bits of the ULPTCR register.

[Table 23.3](#) shows the operation of the ULPTEEn pin in count start or count restart mode.

**Table 23.3 Operation of the ULPTEEn pin in count start mode or count restart mode**

TEECTL[1:0]	ULPTCR.TSTART	ULPTCR.TCSTF	Function of the ULPTEEn pin and counter operation
10 (count start mode)	0 (stop the counter)	—	The edge input at the ULPTEEn pin is invalid. The counter remains stopped.
	1 (start the counter)	0 (counter stopped)	The counter starts counting from the reload value on the edge input at the ULPTEEn pin.
	1 (start the counter)	1 (counter running)	The edge input at the ULPTEEn pin is invalid. The counter continues to count.
11 (count restart mode)	0 (stop the counter)	—	The edge input at the ULPTEEn pin is invalid. The counter remains stopped.
	1 (start the counter)	0 (counter stopped)	The counter starts counting from the reload value on the edge input at the ULPTEEn pin.
	1 (start the counter)	1 (counter running)	The counter is reset on the edge input at the ULPTEEn pin, and it starts counting again from the reload value.

Note: The edge polarity of the ULPTEEn pin can be selected by the TEEPOL[1:0] bits.

### 23.3.8 ULPTIOC : ULPT I/O Control Register

Base address: ULPTn = 0x4022\_0000 + 0x0100 × n (n = 0, 1)  
 ULPTn\_NS = 0x5022\_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TIOGT 0	TIPF[1:0]		—	TOE	—	—

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	TOE	ULPTOn Output Enable 0: Disable the ULPTOn output. 1: Enable the ULPTOn output.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TIPF[1:0]	ULPTEVIn Input Filter*2 0 0: No filter 0 1: Filter sampling at PCLKB 1 0: Filter sampling at PCLKB/8 1 1: Filter sampling at PCLKB/32 This bit is valid only in event counter mode (ULPTMR1.TMOD1 = 1).	R/W
6	TIOGT0	ULPTEVIn Count Control*1 0: Always count external events. 1: Count external events while the ULPTEVIn pin is valid. This bit is valid only in event counter mode (ULPTMR1.TMOD1 = 1) and count enable mode (ULPTMR3.TEECTL[1:0] = 00).	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).

Note 1. The polarity for counting external events can be selected by the ULPTISR.RCCPSEL2 bit.

Note 2. Set the TIPF[1:0] bits to 00 before entering Software Standby mode or Deep Software Standby mode1.

The ULPTIOC register controls the input/output of the ULPT.

The ULPTIOC register is only available for 8-bit read/write access.

#### TIPF[1:0] bits (ULPTEVIn Input Filter)

These bits are used to specify the sampling frequency of the ULPTEVIn input filter. Input to the ULPTEVIn pin is sampled. When a value matches three consecutive times, the value is regarded as the input value.

### 23.3.9 ULPTISR : ULPT Event Pin Select Register

Base address: ULPTn = 0x4022\_0000 + 0x0100 × n (n = 0, 1)  
 ULPTn\_NS = 0x5022\_0000 + 0x0100 × n (n = 0, 1)

Offset address: 0x11

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	RCCP SEL2	—	—

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W



Bit	Symbol	Function	R/W
2	RCCPSEL2	ULPTEEn Polarity Select 0: Count external events when low level. 1: Count external events when high level. This bit is valid only in event counter mode (ULPTMR1.TMOD1 = 1), count enable mode (ULPTMR3.TEECTL[1:0] = 00), and when ULPTIOC.TIOGT0 = 1.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).

The ULPTISR register sets the polarity of the ULPTEEn pin.

The ULPTISR register is only available for 8-bit read/write access.

### 23.3.10 ULPTCMSR : ULPT Compare Match Function Select Register

Base address:  $ULPTn = 0x4022\_0000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $ULPTn\_NS = 0x5022\_0000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x12

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	TOPO LB	TOEB	TCME B	—	TOPO LA	TOEA	TCME A
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TCMEA	Compare Match A Register Enable* <sup>1</sup> 0: Disable compare match A register. 1: Enable compare match A register.	R/W
1	TOEA	ULPTOAn Output Enable 0: Disable the ULPTOAn output. 1: Enable the ULPTOAn output.	R/W
2	TOPOLA	ULPTOAn Polarity Select* <sup>2</sup> 0: Start the ULPTOAn output with low. 1: Start the ULPTOAn output with high.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	TCMEB	Compare Match B Register Enable* <sup>1</sup> 0: Disable compare match B register. 1: Enable compare match B register.	R/W
5	TOEB	ULPTOBn Output Enable 0: Disable the ULPTOBn output. 1: Enable the ULPTOBn output.	R/W
6	TOPOLB	ULPTOBn Polarity Select* <sup>2</sup> 0: Start the ULPTOBn output with low. 1: Start the ULPTOBn output with high.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Do not rewrite this register while the counter is running. Rewrite this register only when the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0).

Note 1. When 1 (stop the counter forcibly) is set to the ULPTCR.TSTOP bit, the TCMEA, and TCMEB bits are set to 0.

Note 2. When data is set to the ULPTMR1 register or 1 is set to the ULPTCR.TSTOP bit, the ULPTOAn and ULPTOBn output is initialized.

The ULPTCMSR register controls the interrupt request signal and input/output of compare match A and compare match B.

The ULPTCMSR register is only available for 8-bit read/write access.

## 23.4 Operation

### 23.4.1 Count operation

The counter is decremented by one on each edge of the count source. When the count value reaches 0x00000000 and the next edge of the count source is input, an interrupt request of underflow is generated. The count operation is set by the ULPTMR1 register and the ULPTMR3 register. [Table 23.4](#) lists the count operations.

**Table 23.4** Count operation

Section No.	Count operation			Count source	ULPTMR1 register		ULPTMR3 register					
	Mode 1	Mode 2	Mode 3		TMOD1	TCK1	TCNTCTL	TEECTL[1:0]				
<a href="#">section 23.4.1.1. Timer &amp; continuous mode</a>	Timer	Continuous	—	ULPTLCLK/ ULPTSCLK	0	0 : ULPTLCLK 1 : ULPTSCLK	0	00				
<a href="#">section 23.4.1.5. Common &amp; continuous &amp; count start mode</a>			Count start					10				
<a href="#">section 23.4.1.6. Common &amp; continuous &amp; count restart mode</a>			Count restart					11				
<a href="#">section 23.4.1.2. Timer &amp; one-shot mode</a>		One-shot	—					ULPTEVIn pin	1	Don't care	0	00
<a href="#">section 23.4.1.7. Common &amp; one-shot &amp; count start mode</a>			Count start									10
<a href="#">section 23.4.1.8. Common &amp; one-shot &amp; count restart mode</a>			Count restart									11
<a href="#">section 23.4.1.3. Event counter &amp; continuous &amp; count enable mode</a>	Event counter	Continuous	Count enable	ULPTEVIn pin	1	Don't care	0					00
<a href="#">section 23.4.1.5. Common &amp; continuous &amp; count start mode</a>			Count start									10
<a href="#">section 23.4.1.6. Common &amp; continuous &amp; count restart mode</a>			Count restart									11
<a href="#">section 23.4.1.4. Event counter &amp; one-shot &amp; count enable mode</a>		One-shot	Count enable					ULPTEVIn pin	1	Don't care	1	00
<a href="#">section 23.4.1.7. Common &amp; one-shot &amp; count start mode</a>			Count start									10
<a href="#">section 23.4.1.8. Common &amp; one-shot &amp; count restart mode</a>			Count restart									11

Note: For details about register settings, see [section 23.3.5. ULPTMR1 : ULPT Mode Register 1](#) and [section 23.3.7. ULPTMR3 : ULPT Mode Register 3](#).

[Mode 1]

The clock used as the count source is selected by the TMOD1 and TCK1 bits of the ULPTMR1 register.

**Timer mode:** The internal clock (ULPTLCLK or ULPTSCLK) is used as the count source.

The counter is decremented on each rising edge of the count source.

**Event counter mode:** The external event input (ULPTEVIn pin) is used as the count source.

The counter is decremented on each edge of the count source.

The edge polarity can be selected by the ULPTMR1.TEDGPL bit and the ULPTMR3.TEVPOL bit.

[Mode 2]

The counter behavior is selected by the TCNTCTL bit of the ULPTMR3 register.

**Continuous mode:** After starting counting, the counter continues to count down (reload value to underflow) repeatedly until the counter stops.

One-shot mode: After starting counting, the counter counts down once (reload value to underflow).

[Mode 3]

The function of the ULPTEEn pin is selected by the TEECTL[1:0] bits of the ULPTMR3 register.

**Count enable mode:** External events are counted while the ULPTEEn pin is valid.

(This bit is valid only in event counter mode.)

**Note:** Because the ULPTEEn pin is synchronized with the count source, external events are counted after three cycles of the count source.

**Count start mode:** The counter starts counting at an edge trigger of the ULPTEEn pin.

The edge polarity of the ULPTEEn pin can be selected by the ULPTMR3.TEEPOL[1:0] bits.

**Note:** Because the ULPTEEn pin is synchronized and held with the count source, external events are counted after four cycles of the count source.

**Count restart mode:** The counter starts counting at the first edge trigger of the ULPTEEn pin.

The counter resets the count at the second or following edge trigger.

The edge polarity of the ULPTEEn pin can be selected by the ULPTMR3.TEEPOL[1:0] bits.

**Note:** Because the ULPTEEn pin is synchronized and held with the count source, external events are counted after four cycles of the count source.

### 23.4.1.1 Timer & continuous mode

This mode operates in timer mode only.

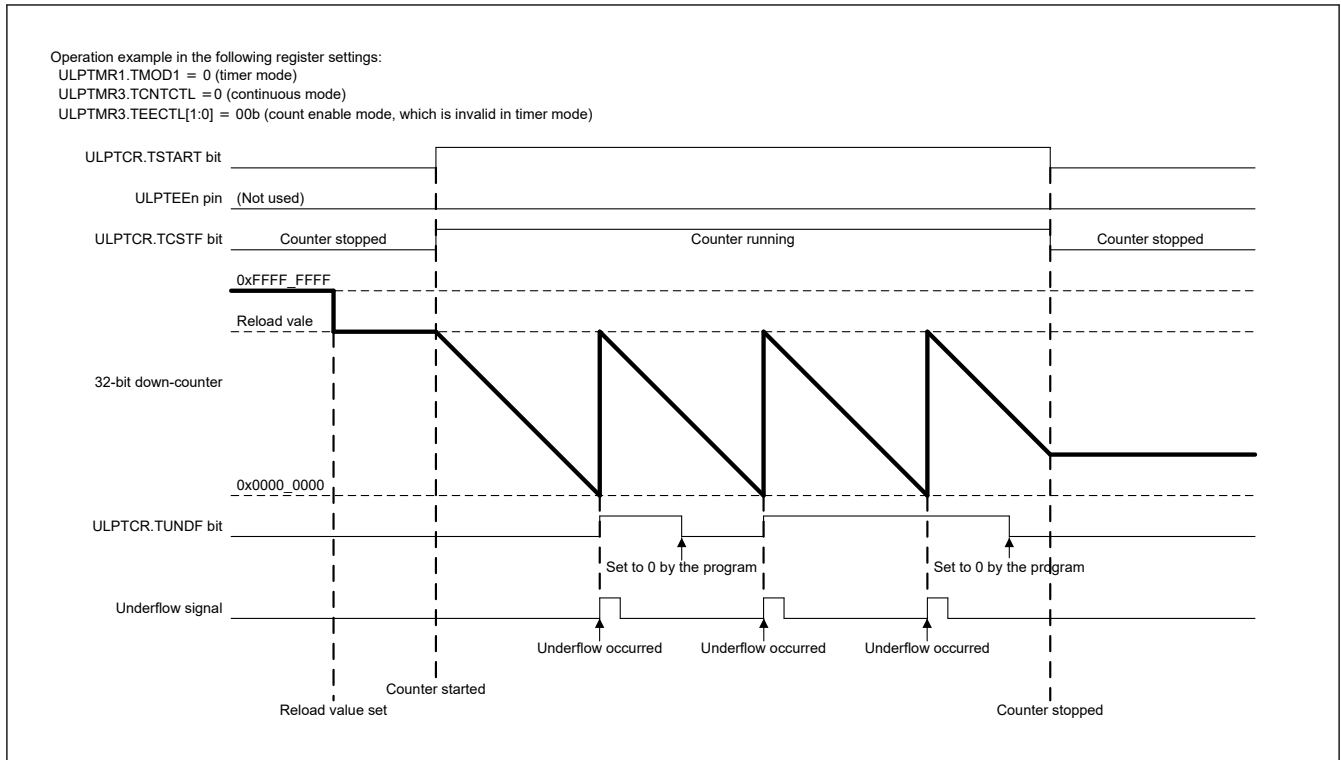
The counter continues to count down (reload value to underflow) repeatedly in synchronization with the count source.

When 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1).

When an underflow occurs, the counter counts down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.2 shows an operation example.



**Figure 23.2** Operation example in timer & continuous mode

### 23.4.1.2 Timer & one-shot mode

This mode operates in timer mode only.

The counter counts down (reload value to underflow) once in synchronization with the count source.

When 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1).

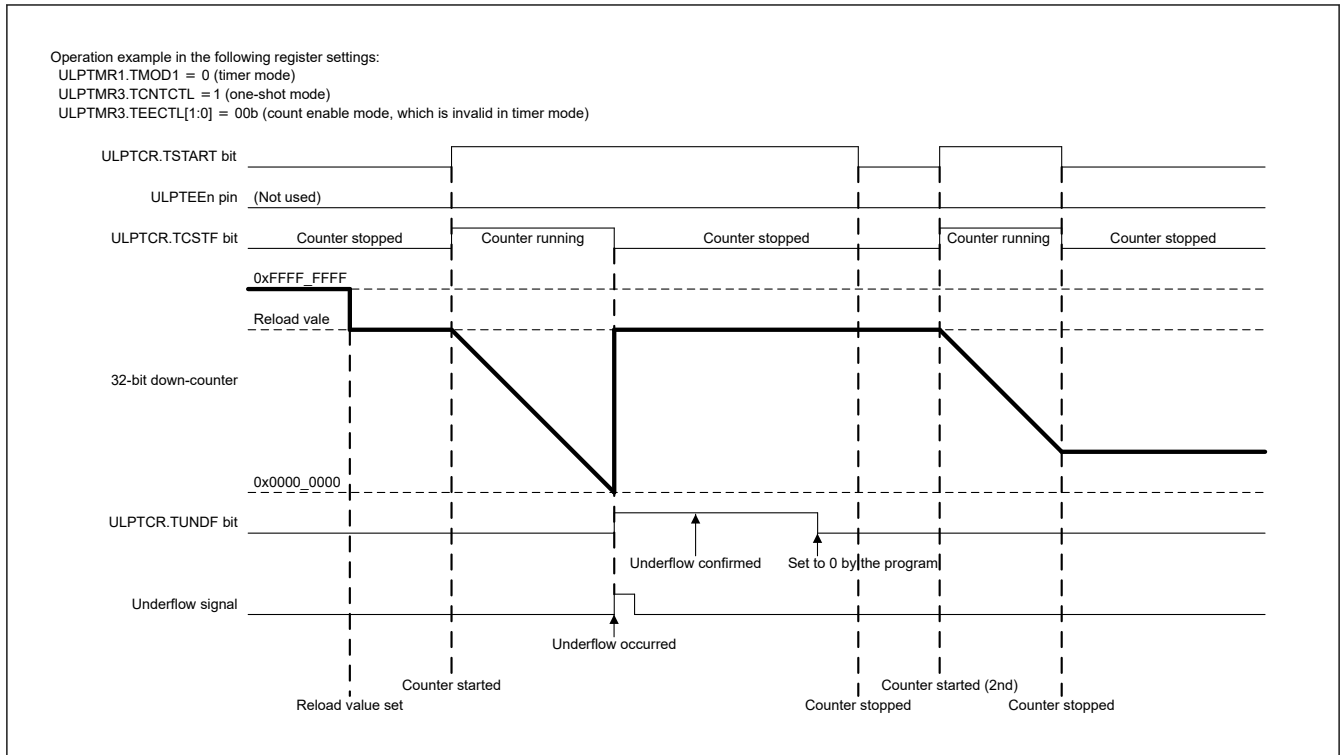
When an underflow occurs, the counter is stopped (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0).

To restart the counter after the counter is stopped, perform the following procedure:

1. Read the ULPTCR.TUNDF bit to confirm that it is set to 1 (underflow occurred). This is to make sure that counting down is finished.
2. Write 0 to the ULPTCR.TSTART bit to stop the counter.
3. Write 1 to the ULPTCR.TSTART bit to start the counter.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.3 shows an operation example.



**Figure 23.3** Operation example in timer & one-shot mode

### 23.4.1.3 Event counter & continuous & count enable mode

This mode operates in event counter mode only.

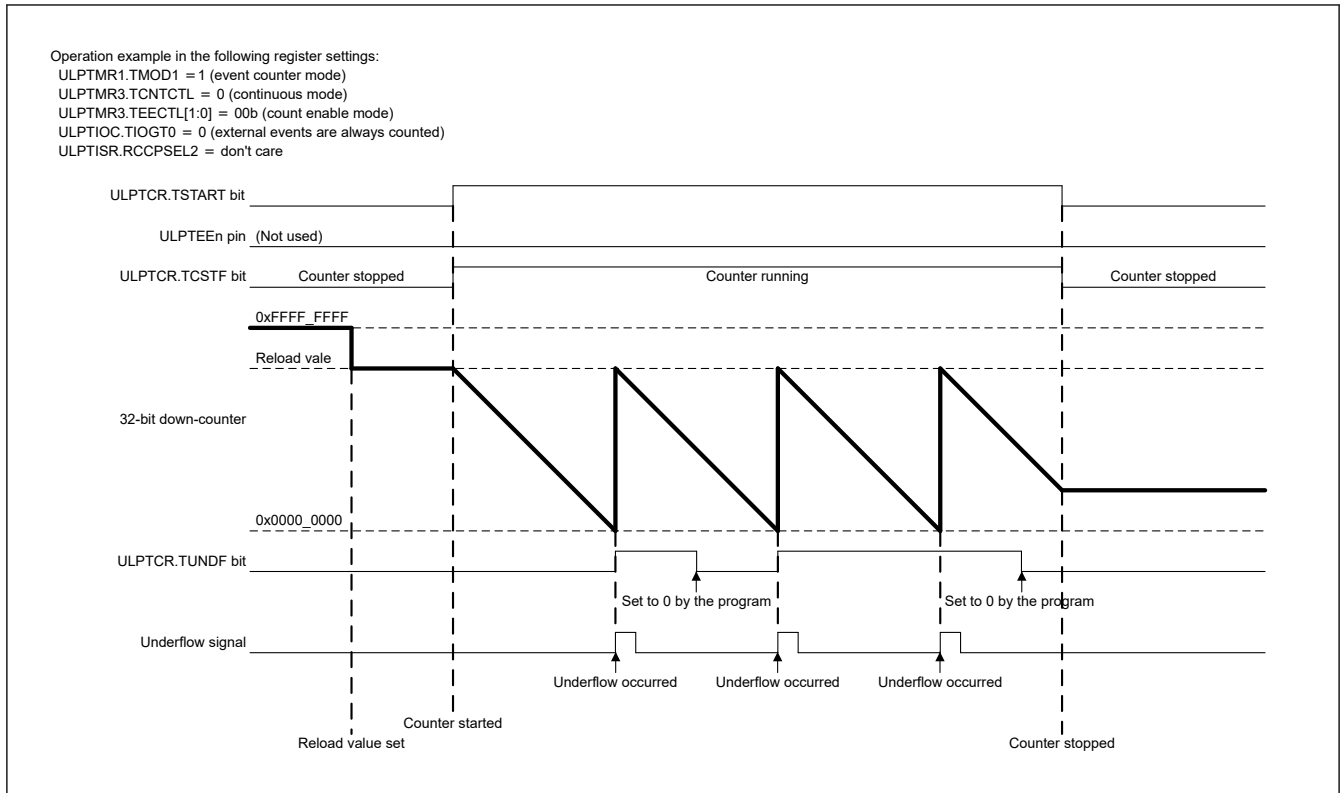
The counter continues to count down (reload value to underflow) repeatedly in synchronization with the count source. The duration in which the count source is counted can be specified by the ULPTIOC.TIOGT0 bit and ULPTISR.RCCPSEL2 bit.

When 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1).

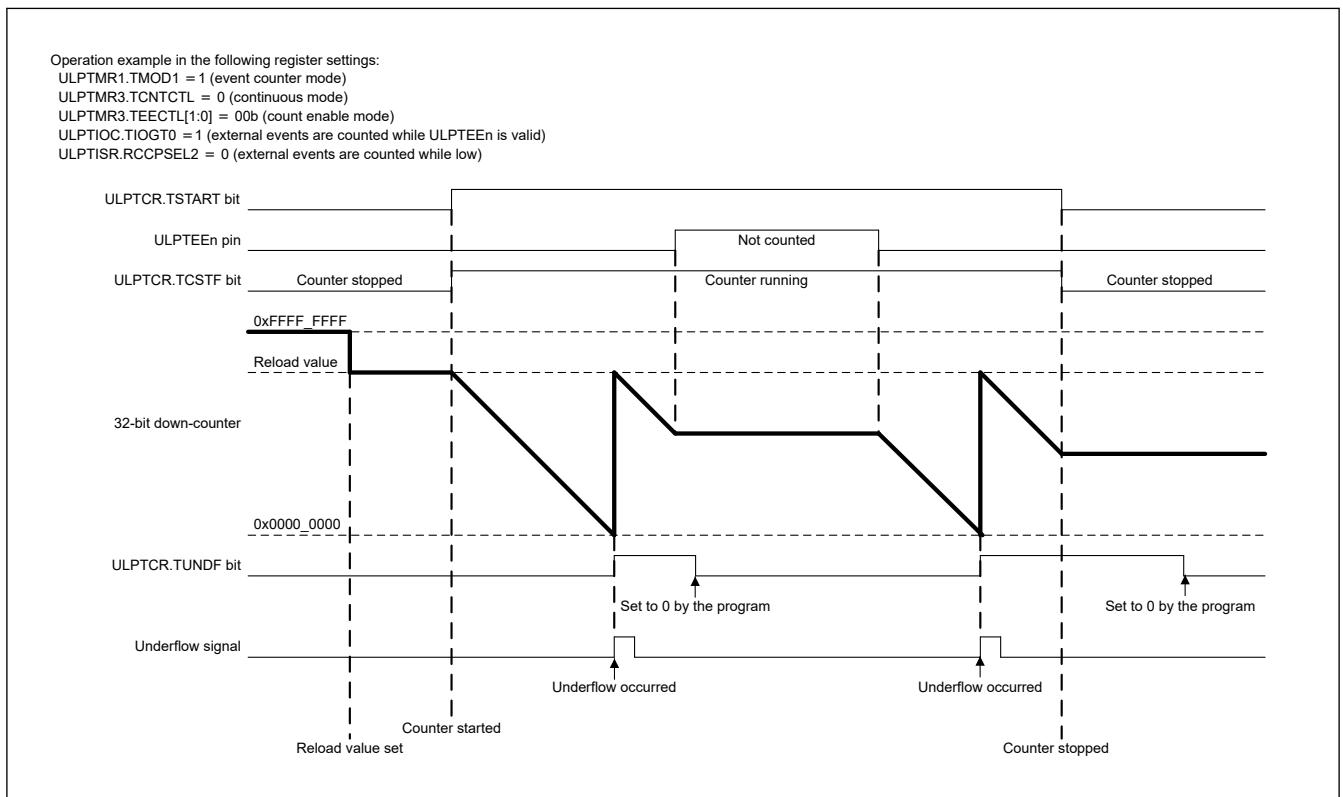
When an underflow occurs, the counter counts down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

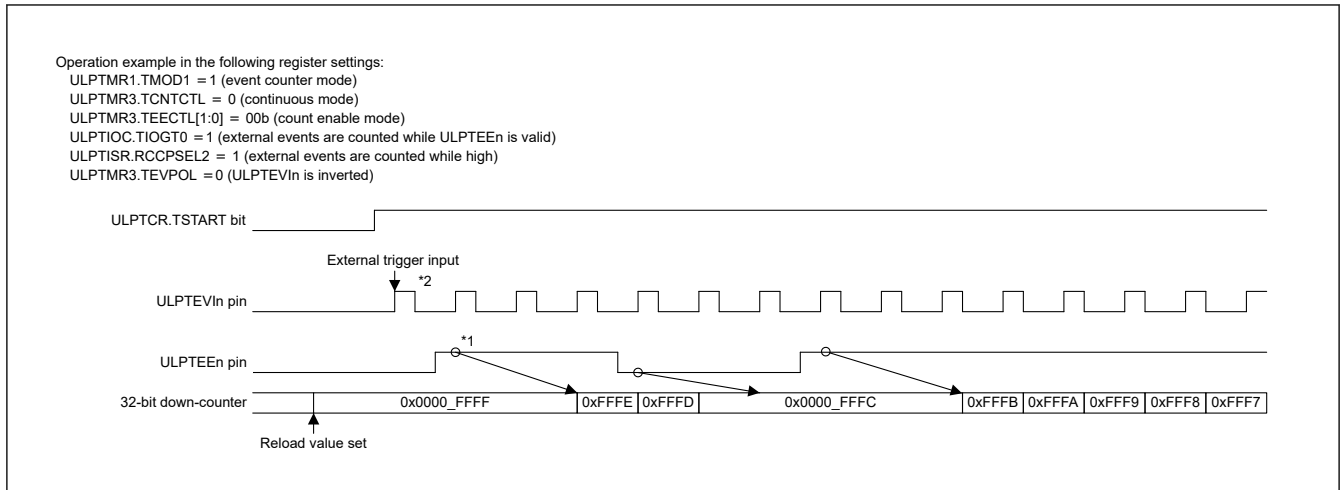
Figure 23.4 shows an operation example when the ULPTIOC.TIOGT0 bit is set to 0, and Figure 23.5 and Figure 23.6 show operation examples when the ULPTIOC.TIOGT0 bit is set to 1.



**Figure 23.4** Operation example in event counter & continuous & count enable mode (with the ULPTIOC.TIOGT0 bit set to 0)



**Figure 23.5** Operation example 1 in event counter & continuous & count enable mode (with the ULPTIOC.TIOGT0 bit set to 1)



**Figure 23.6 Operation example 2 in event counter & continuous & count enable mode (with the ULPTIOC.TIOGT0 bit set to 1)**

Note 1. When synchronization is controlled, there is a delay by two cycles of count source until external events are actually counted. In addition, the count start timing may be shifted by one cycle due to the phase difference between the ULPTEEn pin and the sampling clock.

Note 2. Depending on the state before the counter is stopped, counting may be performed during two cycles of the count source immediately after the counter is started.

To disable the counting operation during two cycles immediately after the counter is started, set 1 to the ULPTCR.TSTOP bit to initialize the internal circuits, and then finish the operation settings before starting the counter.

#### 23.4.1.4 Event counter & one-shot & count enable mode

This mode operates in event counter mode only.

The counter counts down (reload value to underflow) once in synchronization with the count source. The duration in which the count source is counted can be specified by the ULPTIOC.TIOGT0 bit and ULPTISR.RCCPSEL2 bit.

When 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1).

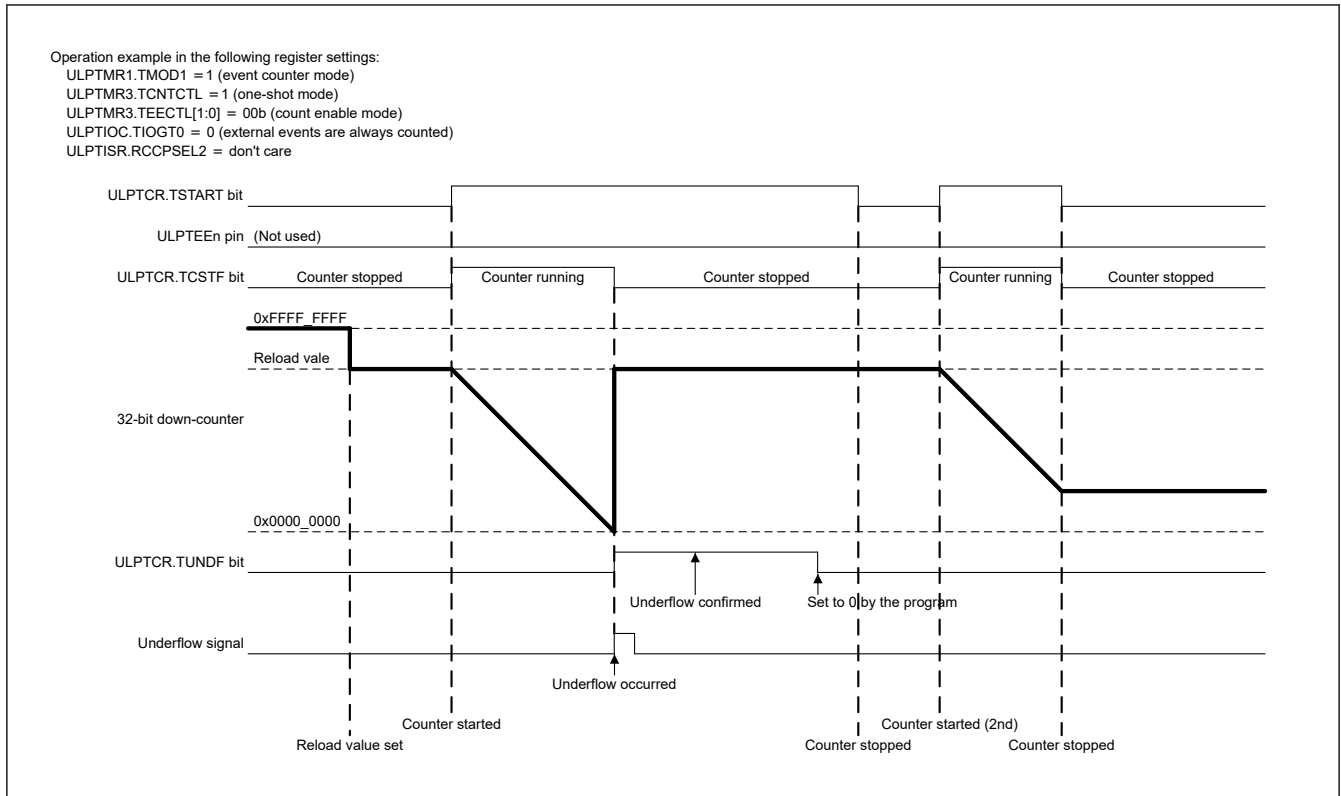
When an underflow occurs, the counter is stopped (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0).

To restart the counter after the counter is stopped, perform the following procedure:

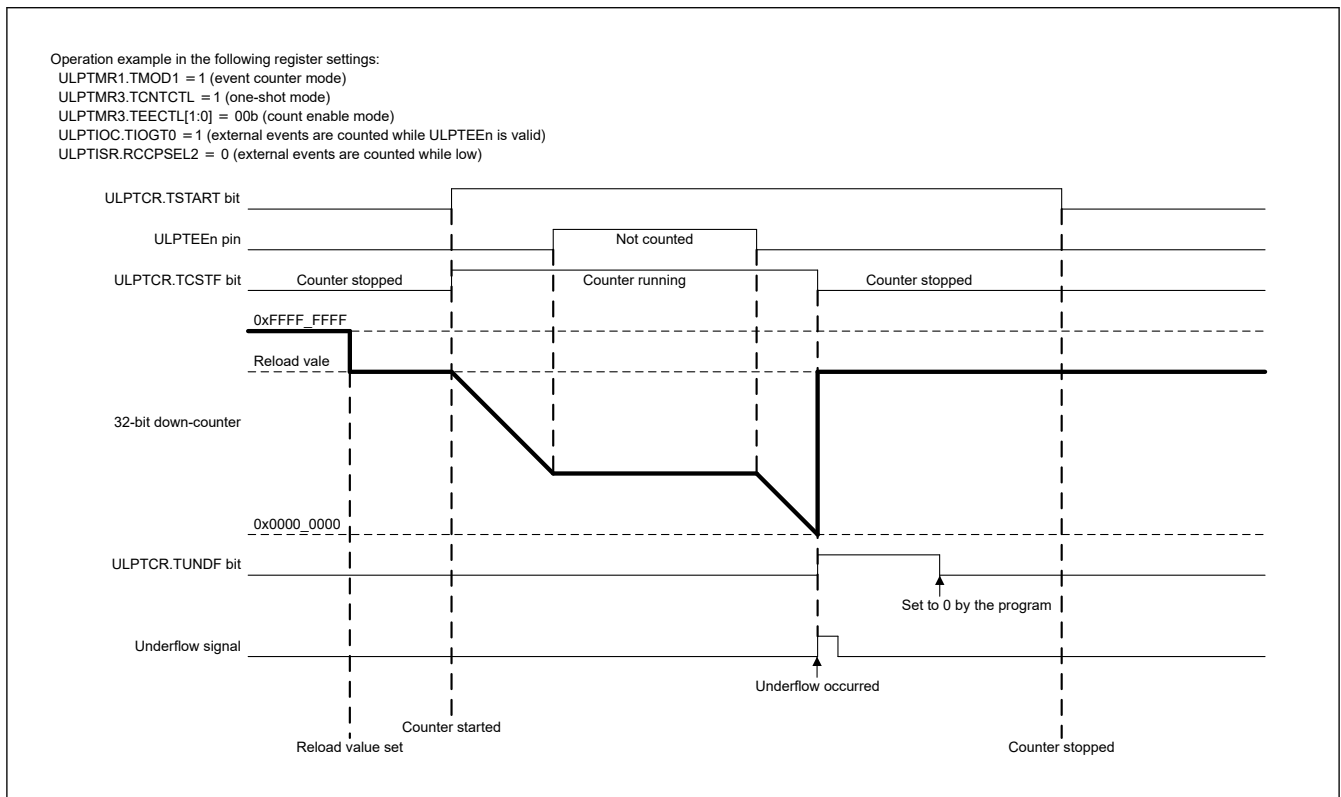
1. Read the ULPTCR.TUNDF bit to confirm that it is set to 1 (underflow occurred). This is to make sure that counting down is finished.
2. Write 0 to the ULPTCR.TSTART bit to stop the counter.
3. Write 1 to the ULPTCR.TSTART bit to start the counter.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.7 shows an operation example when the ULPTIOC.TIOGT0 bit is set to 00, and Figure 23.8 shows an operation example when the ULPTIOC.TIOGT0 bit is set to 01.



**Figure 23.7** Operation example in event counter & one-shot & count enable mode (with the ULPTIOC.TIOGT0 bit set to 0)



**Figure 23.8** Operation example in event counter & one-shot & count enable mode (with the ULPTIOC.TIOGT0 bit set to 1)



### 23.4.1.5 Common & continuous & count start mode

This mode operates in both timer mode and event counter mode.

The counter continues to count down (reload value to underflow) repeatedly in synchronization with the count source.

After 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1) by an edge trigger at the ULPTEEn pin.

When an underflow occurs, the counter counts down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.9 shows an operation example.

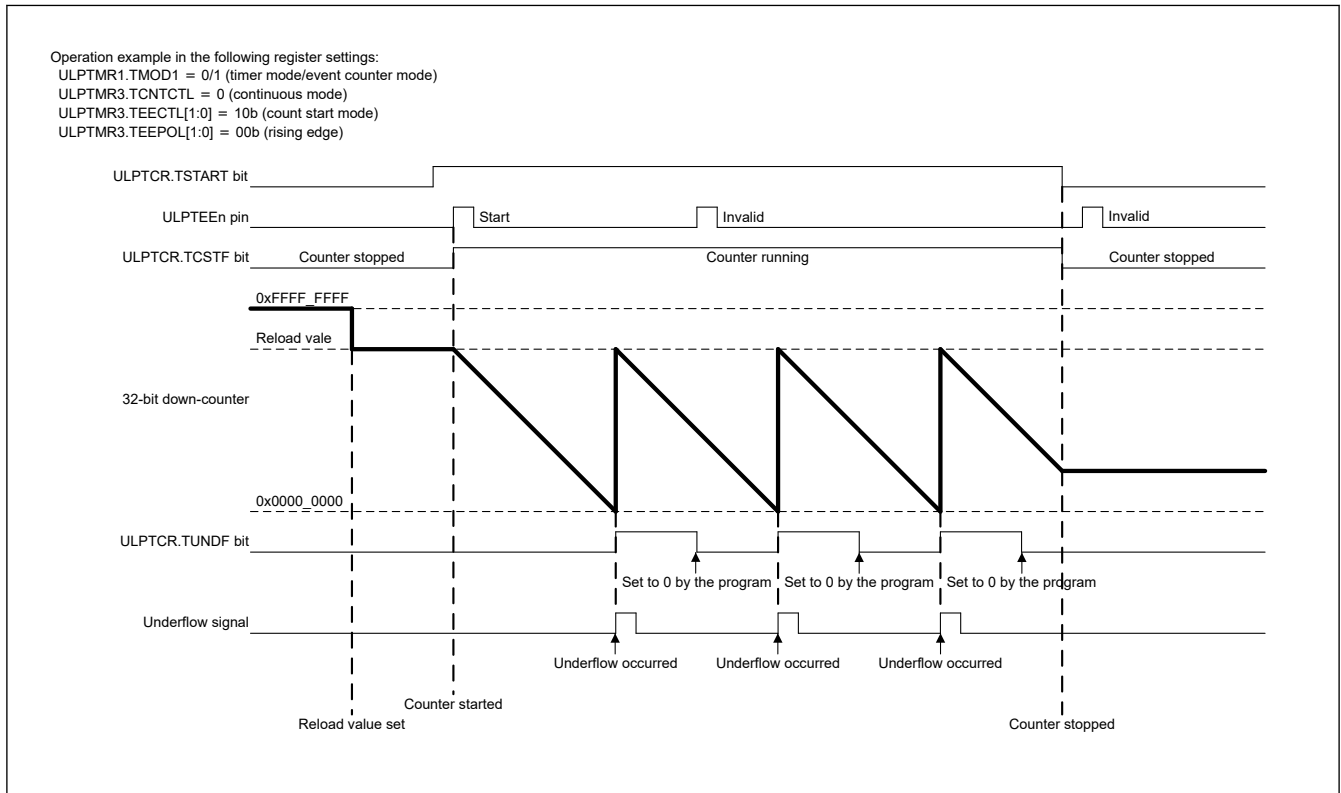


Figure 23.9 Operation example in common & continuous & count start mode

### 23.4.1.6 Common & continuous & count restart mode

This mode operates in both timer mode and event counter mode.

The counter continues to count down (reload value to underflow) repeatedly in synchronization with the count source.

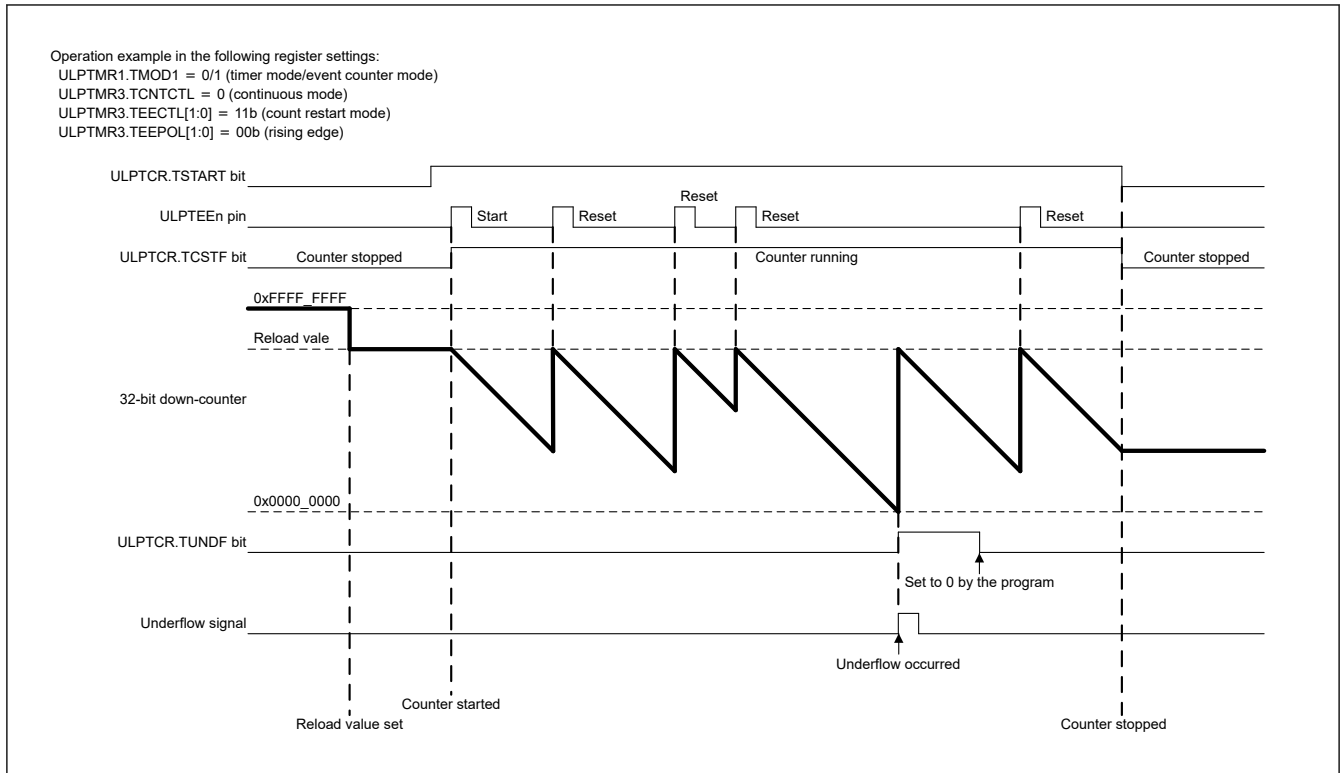
After 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1) by an edge trigger at the ULPTEEn pin.

On an edge trigger input at the ULPTEEn pin when the counter is running (with ULPTCR.TCSTF bit set to 1), the counter is reset and it starts counting again from the reload value.

If no edge trigger is input from the ULPTEEn pin and an overflow occurs, the counter starts counting again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.10 shows an operation example.



**Figure 23.10 Operation example in common & continuous & count restart mode**

### 23.4.1.7 Common & one-shot & count start mode

This mode operates in both timer mode and event counter mode.

The counter counts down (reload value to underflow) once in synchronization with the count source.

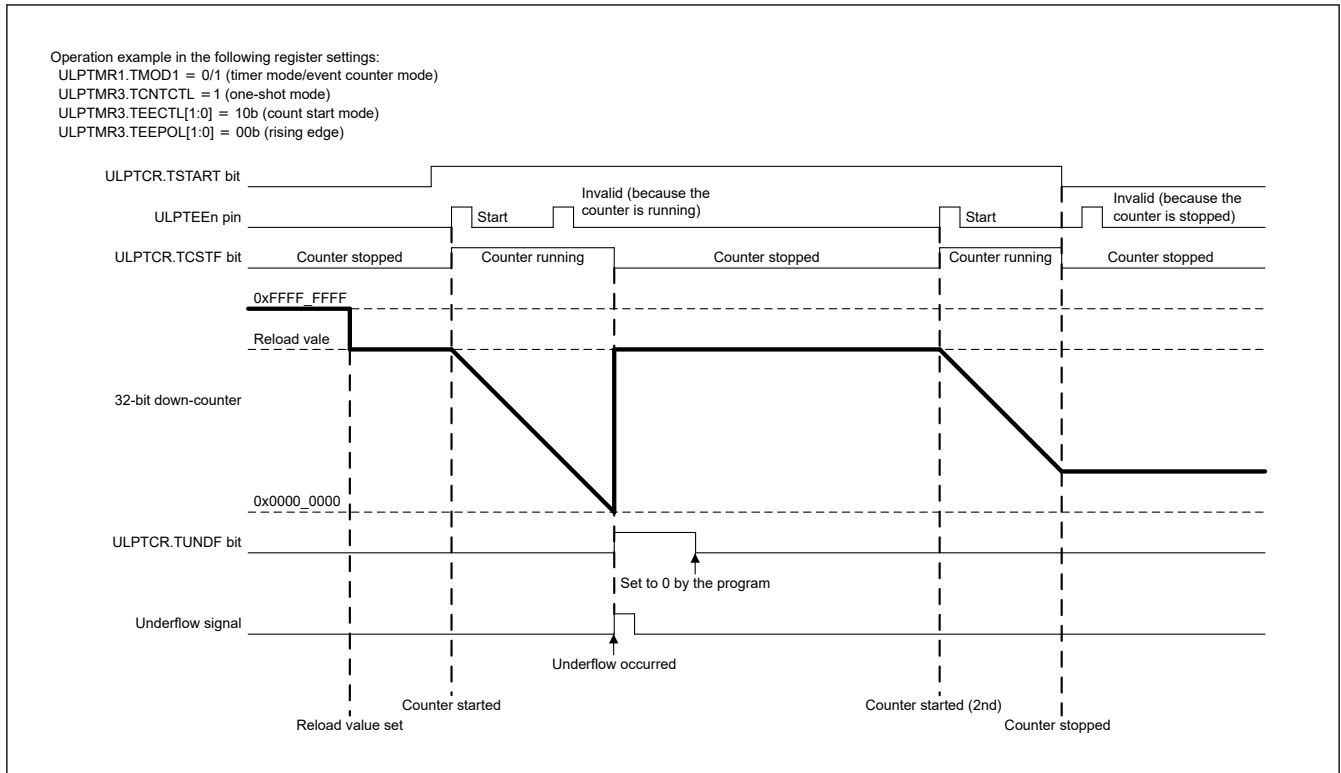
After 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1) by an edge trigger at the ULPTEEn pin.

When an underflow occurs, the counter is stopped (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0).

On an edge trigger input at the ULPTEEn pin when the counter is running (with ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0), the counter starts counting down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.11 shows an operation example.



**Figure 23.11 Operation example in common & one-shot & count start mode**

### 23.4.1.8 Common & one-shot & count restart mode

This mode operates in both timer mode and event counter mode.

The counter counts down (reload value to underflow) once in synchronization with the count source.

After 1 is set to the ULPTCR.TSTART bit, the counter is started (with the ULPTCR.TCSTF bit set to 1) by an edge trigger of the ULPTEEn pin.

On an edge trigger input at the ULPTEEn pin when the counter is running (with ULPTCR.TCSTF bit set to 1), the counter is reset and it starts counting again from the reload value.

If no edge trigger is input at the ULPTEEn pin and an underflow occurs, the counter is stopped (with the ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0).

On an edge trigger input at the ULPTEEn pin when the counter is running (with ULPTCR.TSTART bit set to 1 and the ULPTCR.TCSTF bit set to 0), the counter starts counting down again from the reload value.

When 0 is set to the ULPTCR.TSTART bit, the counter is stopped (with the ULPTCR.TCSTF bit set to 0).

Figure 23.12 shows an operation example.

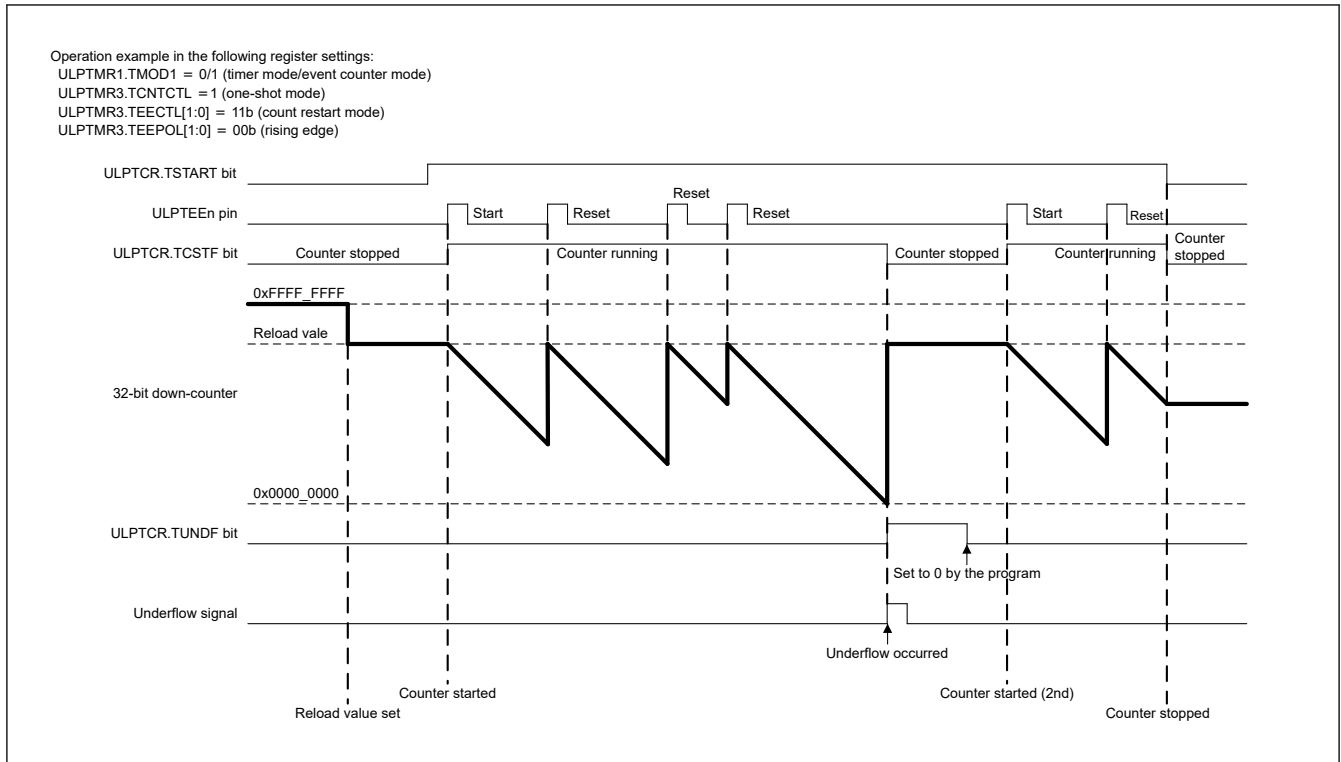


Figure 23.12 Operation example in common & one-shot & count restart mode

### 23.4.2 Rewriting the counter and reload register

Regardless of the operating mode, the counter and reload register rewrite timing changes depending on the TSTART and TCSTF bits of the ULPTCR register, the TCMEA and TCMEB bits of the ULPTCMSR register, and the TEECTL[1:0] bits of the ULPTMR3 register.

**Condition A: When the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0)**

A reload value is written directly to the counter and the reload register.

**Condition B: When the counter is running (with the ULPTCR.TCSTF bit set to 1) and the compare match A/B registers are disabled (with both the TCMEA and TCMEB bits of the ULPTCR register set to 0)**

After a reload value is written to the ULPTCNT register, the reload value is set first to the reload register, then to the counter in synchronization with the count source.

**Condition C: When the counter is running (with the ULPTCR.TCSTF bit set to 1) and the compare match A/B registers are enabled (with either the TCMEA bit or the TCMEB bit of the ULPTCR register set to 1)**

After a reload value is written to the ULPTCNT register, the reload value is written to the reload register in synchronization with the count source. Then, the reload value is set to the counter in synchronization with an underflow.

**Condition D: When the counter is running (with ULPTCR.TCSTF bit set to 1) and when the count restart mode is enabled (with the ULPTMR3.TEECTL[1:0] bits set to 11)**

After an edge trigger is an input from the ULPTEEn pin, the reload value set in the reload register is set to the counter in synchronization with the count source. In this condition, the reload register is not rewritten.

Figure 23.13 to Figure 23.15 show the counter and reload register rewrite timing.

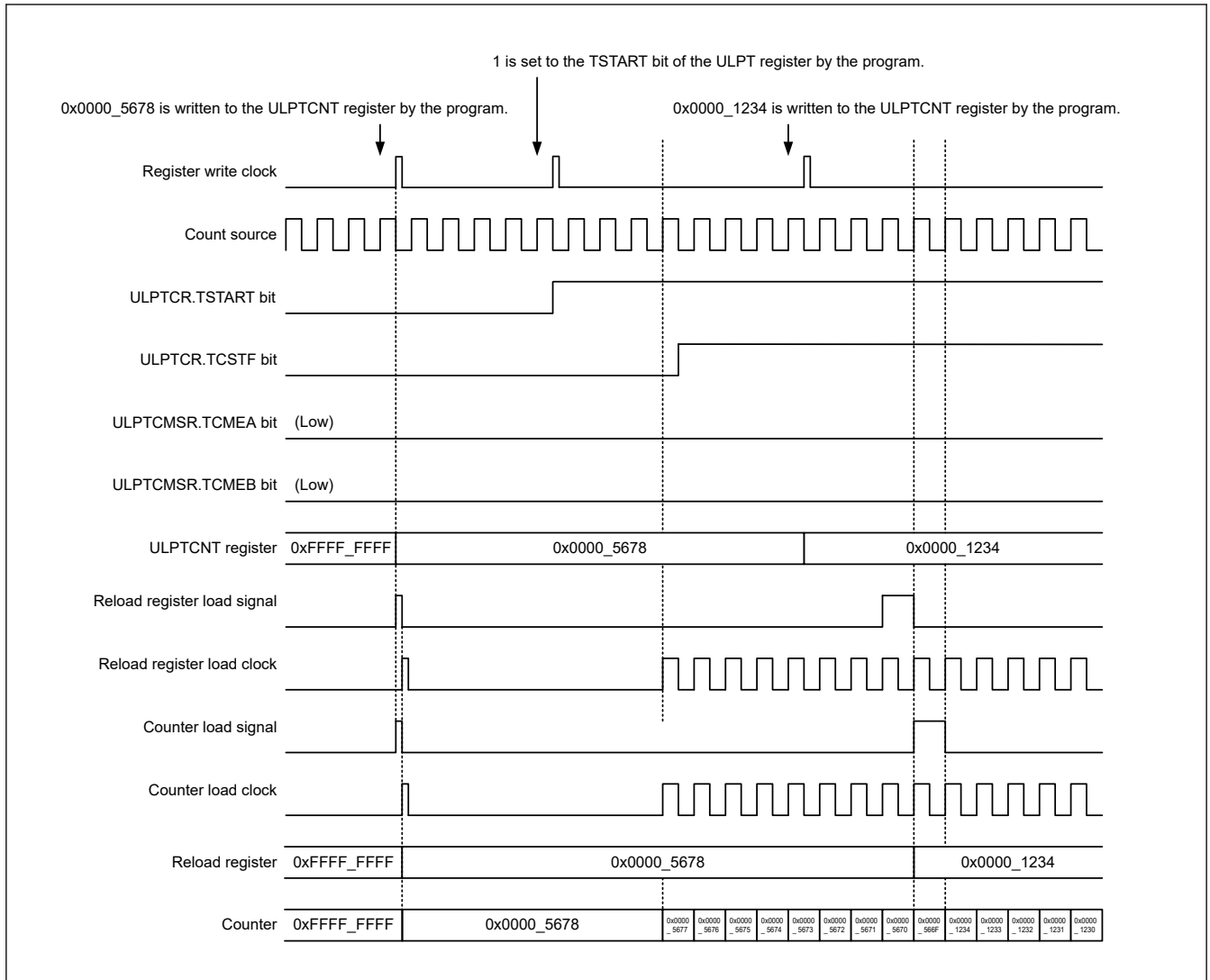


Figure 23.13 Counter and reload rewrite timing (under condition A or B)

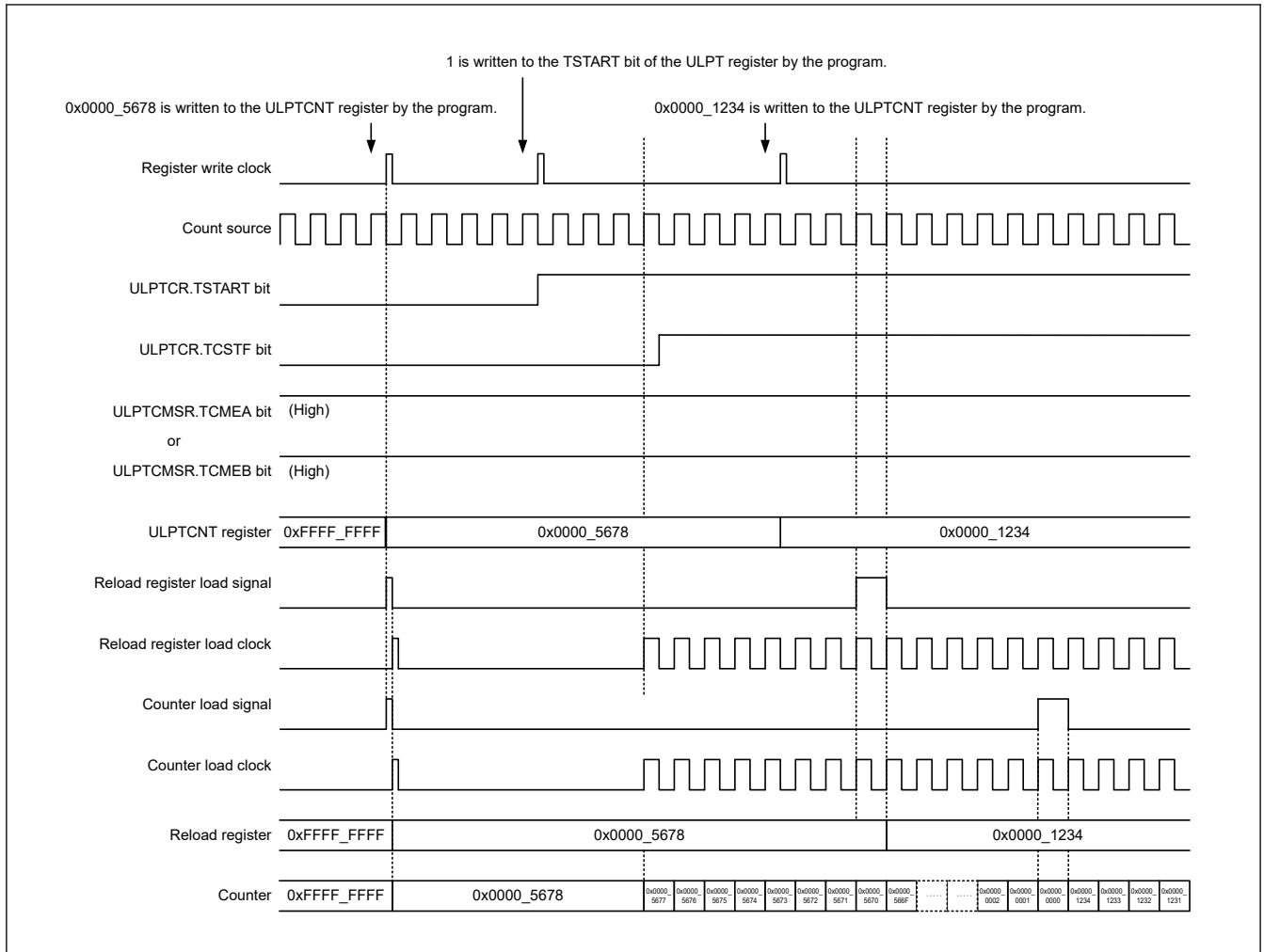


Figure 23.14 Counter and reload rewrite timing (under condition A or C)

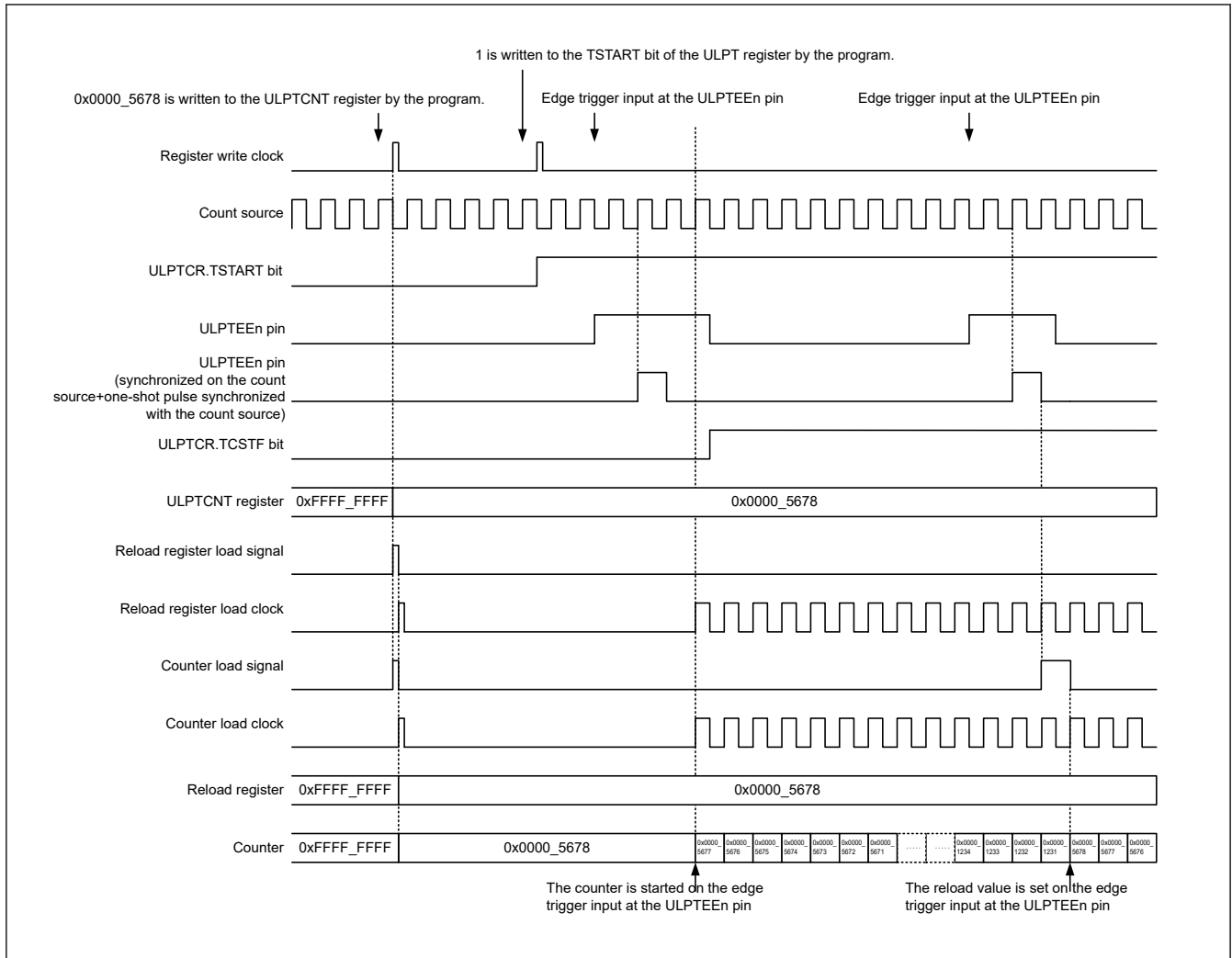


Figure 23.15 Counter and reload rewrite timing (under condition D)

### 23.4.3 Rewriting the compare circuits and reload registers for compare match A/B

Regardless of the operating mode, the rewrite timing for the compare circuits and reload registers for compare match A/B changes depending on the TSTART and TCSTF bits of the ULPTCR register.

**Condition A: When the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0)**

Compare match A/B values are written directly to the compare circuits and reload registers.

**Condition B: When the counter is running (with the ULPTCR.TCSTF bit set to 1)**

After compare match A/B values are written to the ULPTCMA and ULPTCMB registers, the compare match A/B values are written to the reload registers in synchronization with the count source. Then, the compare match A/B values are written to the compare circuits in synchronization with an underflow.

Figure 23.16 shows the rewrite timing for the compare circuit and reload register for compare match A.

The timing for compare match B is similar to that for compare match A.

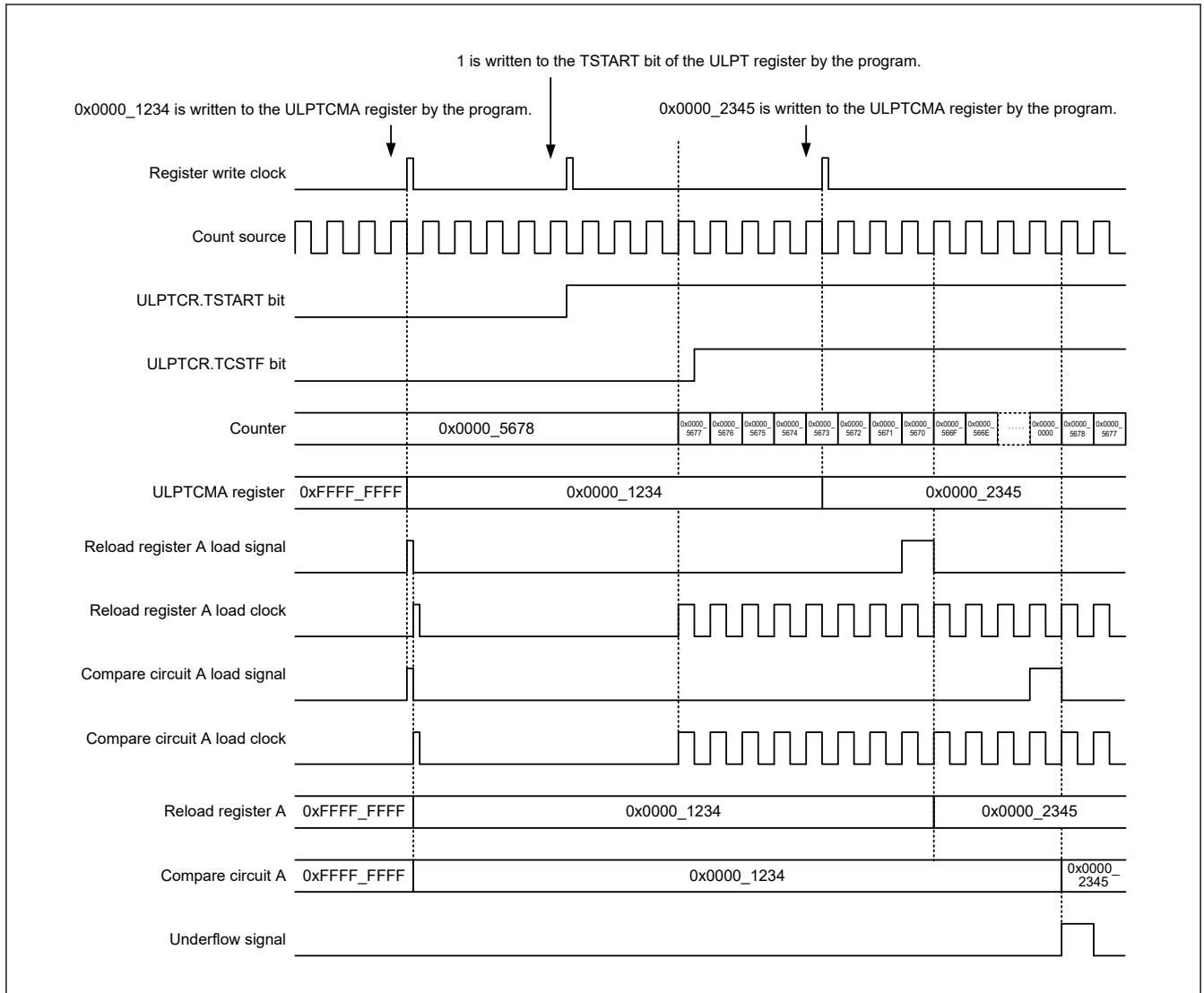


Figure 23.16 Rewrite timing for the compare circuit and reload register for compare match A

### 23.4.4 Pulse output

Regardless of the operating mode, the ULPTOn pin can output pulses. The output level is inverted each time the counter underflows. The pulse output from the ULPTOn pin can be stopped by the ULPTIOC.TOE bit. The initial value of the output level can be selected by the ULPTMR3.TOPOL bit.

Figure 23.17 shows an operation example of pulse output.



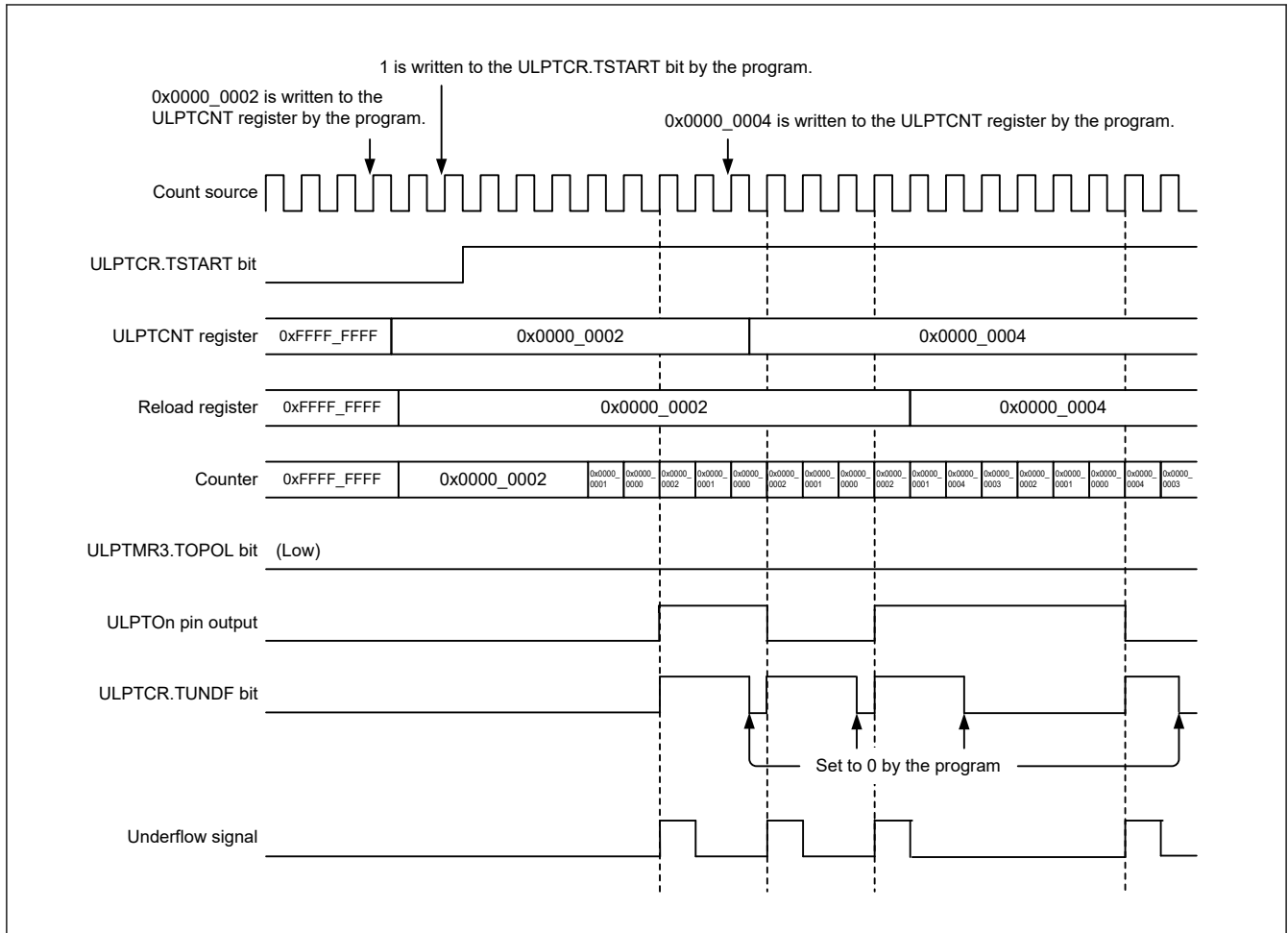


Figure 23.17 Operation example of pulse output

### 23.4.5 Compare match function

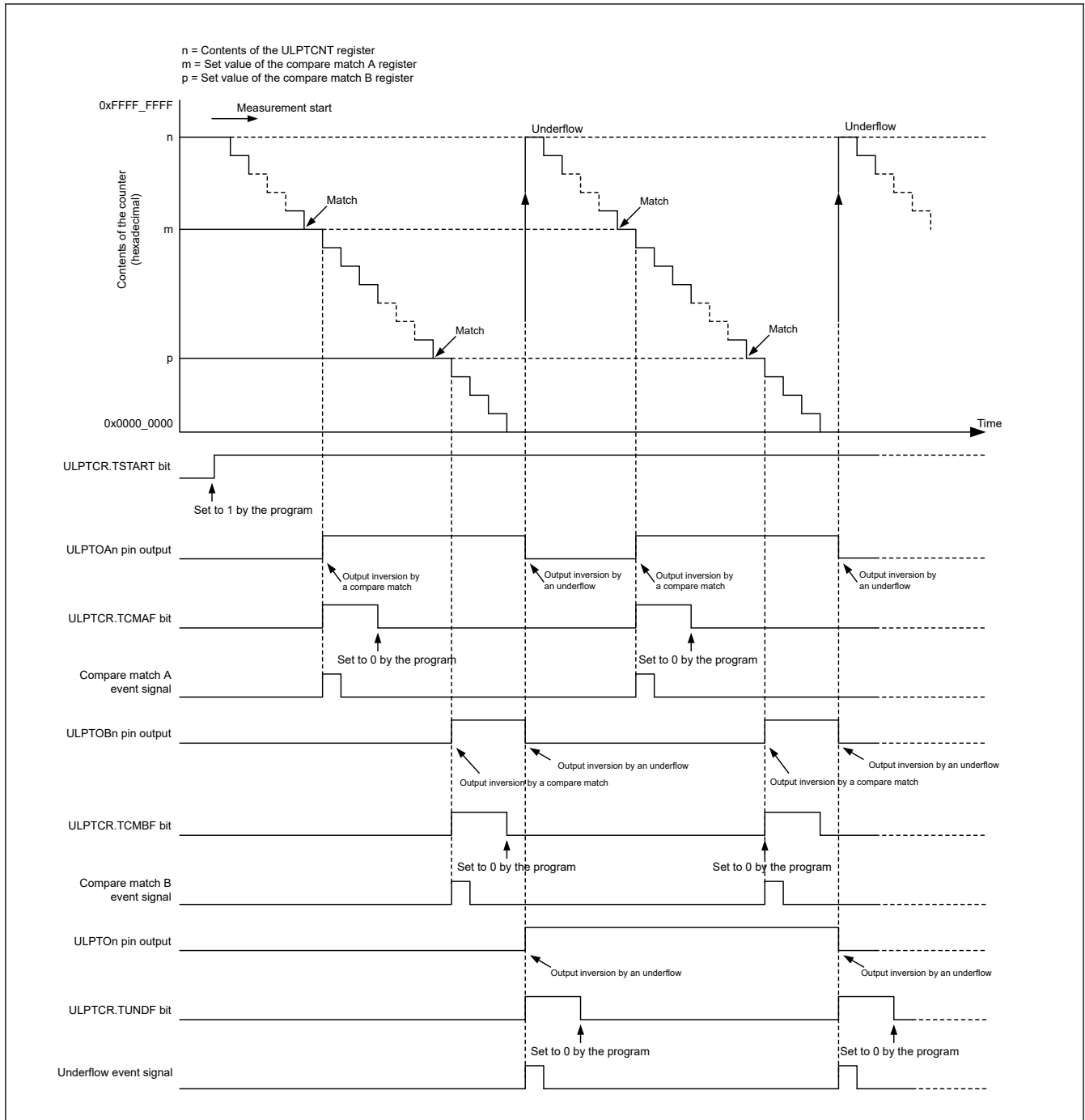
This function detects a match (compare match) between the ULPTCMA/ULPTCMB register contents and the ULPTCNT register contents. This function is enabled when the TCMEA or TCMEB bit of the ULPTCMSR register is set to 1 (the compare match A/B registers are enabled).

The counter is decremented in synchronization with the count source selected by the TMOD1/TCK1 bits of the ULPTMR1 register. When the ULPTCNT register value matches the ULPTCMA or ULPTCMB register value, the TCMAF or TCMBF bit of the ULPTCR register is set to 1 and an interrupt request is generated.

The counter and reload register rewrite timing differs depending on whether the compare match function is enabled. For details, see [section 23.4.2. Rewriting the counter and reload register](#).

The output level of the ULPTOAn/ULPTOBn pin is inverted upon a match or underflow. The output level can be selected by the ULPTOAn/ULPTOBn bit of the ULPTCMSR register.

Figure 23.18 shows an operation example of the compare match function.



**Figure 23.18** Operation example of the compare match function (both the TOPOLA and TOPOLB bits of the ULPTCMSR register are set to 0)

### 23.4.6 Input and output settings for each mode

Table 23.5 to Table 23.9 show the states of the input pins (ULPTEEn and ULPTEVIn) and output pins (ULPTOn, ULPTOAn, and ULPTOBn) in each mode.

**Table 23.5** ULPTEEn pin settings

Operating mode		ULPTMR1 register	ULPTMR3 register	ULPTEEn pin input
		TMOD1 bit	TEECTL[1:0] bit	
Timer mode	-	0	00	Input disabled
	Count start		10	Normal input
	Count restart		11	
Event counter mode	Count enable	1	00	
	Count start		10	
	Count restart		11	

**Table 23.6** ULPTEVIn pin settings

Operating mode	ULPTMR1 register	ULPTMR3 register	ULPTEVIn pin input
	TMOD1 bit	TEVPOL bit	
Timer mode	0	0 or 1	Input disabled
Event counter mode	1	0	Normal input
		1	Inverted input

**Table 23.7** ULPTOn pin settings

Operating mode	ULPTIOC register	ULPTMR3 register	ULPTOn pin output
	TOE bit	TOPOL bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

**Table 23.8** ULPTOAn pin settings

Operating mode	ULPTCMSR register		ULPTOAn pin output
	TOEA bit	TOPOLA bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

**Table 23.9** ULPTOBn pin settings

Operating mode	ULPTCMSR register		ULPTOBn pin output
	TOEB bit	TOPOLB bit	
All modes	1	1	Inverted output
		0	Normal output
	0	0 or 1	Output disabled

### 23.4.7 Standby mode

The ULPT can operate in Software Standby mode or Deep Software Standby mode<sup>1</sup>. Set it to each Standby mode with count operation start (ULPTCR.TSTART = 1, ULPTCR.TCSTF = 1).

[Table 23.10](#) lists the settings available in Software Standby mode or Deep Software Standby mode<sup>1</sup>.

**Table 23.10 Settings available in Software Standby mode or Deep Software Standby mode1**

Section No.	Count operation			Count source	CPU recovery source
	Mode 1	Mode 2	Mode 3		
<a href="#">section 23.4.1.1. Timer &amp; continuous mode</a>	Timer	Continuous	—	ULPTLCLK, ULPTSCLK	<ul style="list-style-type: none"> <li>• Underflow</li> <li>• Compare match A/B</li> </ul>
<a href="#">section 23.4.1.5. Common &amp; continuous &amp; count start mode</a>			Count start		
<a href="#">section 23.4.1.6. Common &amp; continuous &amp; count restart mode</a>			Count restart		
<a href="#">section 23.4.1.2. Timer &amp; one-shot mode</a>		One-shot	—		
<a href="#">section 23.4.1.7. Common &amp; one-shot &amp; count start mode</a>			Count start		
<a href="#">section 23.4.1.8. Common &amp; one-shot &amp; count restart mode</a>			Count restart		
<a href="#">section 23.4.1.3. Event counter &amp; continuous &amp; count enable mode</a>	Event counter	Continuous	Count enable	ULPTEVIn pin <sup>*1</sup>	<ul style="list-style-type: none"> <li>• Underflow</li> <li>• Compare match A/B</li> </ul>
<a href="#">section 23.4.1.5. Common &amp; continuous &amp; count start mode</a>			Count start		
<a href="#">section 23.4.1.6. Common &amp; continuous &amp; count restart mode</a>			Count restart		
<a href="#">section 23.4.1.4. Event counter &amp; one-shot &amp; count enable mode</a>		One-shot	Count enable		
<a href="#">section 23.4.1.7. Common &amp; one-shot &amp; count start mode</a>			Count start		
<a href="#">section 23.4.1.8. Common &amp; one-shot &amp; count restart mode</a>			Count restart		

Note: The ULPT can operate in all modes even in Software Standby mode or Deep Software Standby mode1. For details about the count operations and register settings, see [section 23.4.1. Count operation](#).

Note 1. Only ULPTEVIn pin can be used in Deep Software Standby mode1.

Only in event counter mode (ULPTMR1.TMOD1 = 1), set the ULPTIOC.TIPF[1:0] bits to 00 (no filter) before entering standby mode.

In Software Standby mode or Deep Software Standby mode1, the digital filter clock (PCLKB) stops (the digital filter does not function).

It is prohibited to rewrite the ULPTCNT, ULPTCMA, and ULPTCMB registers immediately before setting each standby mode.

If the ULPTCNT, ULPTCMA, and ULPTCMB registers are rewritten while the counter is running, set each standby mode after four or more cycles of the count source.

### 23.4.8 Interrupt sources

The ULPT has three interrupt sources: ULPTI, ULPTCMAI, and ULPTCMBI.

[Table 23.11](#) lists the interrupt sources.

**Table 23.11 ULPT interrupt sources**

Name	Interrupt source	DMAC/DTC activation
ULPTn_ULPTI	When the counter underflows	Possible
ULPTn_ULPTCMAI	When the ULPTCNT value matches the ULPTCMA value	Possible
ULPTn_ULPTCMBI	When the ULPTCNT value matches the ULPTCMB value	Possible

### 23.4.9 Event output to the ELC

The ULPT is capable of link operation for a specified module via the event link controller (ELC) by using the interrupt request signal as an event signal.

The ULPT outputs an event signal when compare match A/compare match B or an underflow occurs. For details, see [section 18, Event Link Controller \(ELC\)](#).

## 23.5 Usage notes

### 23.5.1 Start and stop control of the counter

**Condition A: In timer mode (ULPTMR1.TMOD1 = 0), continuous mode (ULPTMR3.TCNTCTL = 0), and count enable mode (ULPTMR3.TEECTL[1:0] = 00)**

When the counter is stopped (with the TCSTF bit of the ULPTCR register set to 0), even if 1 (start the counter) is set to the TSTART bit of the ULPTCR register, the TCSTF bit remains at 0 (counter stopped) for five cycles of the count source.

Do not access the registers<sup>\*1</sup> related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 1 (counter running).

When the counter is running (with the TCSTF bit set to 1), even if 0 (stop the counter) is set to the TSTART bit, the TCSTF bit remains at 1 (counter running) for five cycles of the count source.

Do not access the registers<sup>\*1</sup> related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 0 (counter stopped).

Before changing the TSTART bit from 0 to 1, clear the interrupt register. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

Note 1. Registers related to the ULPT: ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, and ULPTCMSR

**Condition B: In event counter mode (ULPTMR1.TMOD1 = 1), continuous mode (ULPTMR3.TCNTCTL = 0), and count enable mode (ULPTMR3.TEECTL[1:0] = 00)**

When the counter is stopped (with the TCSTF bit of the ULPTCR register set to 0), even if 1 (start the counter) is set to the TSTART bit of the ULPTCR register, the TCSTF bit remains at 0 (counter stopped) for two cycles of the count source.

Do not access the registers<sup>\*1</sup> related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 1 (counter running).

When the counter is running (with the TCSTF bit set to 1), even if 0 (stop the counter) is set to the TSTART bit, the TCSTF bit remains at 1 (counter running) for two cycles of the count source.

Do not access the registers<sup>\*1</sup> related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 0 (counter stopped).

Before changing the TSTART bit from 0 to 1, clear the interrupt register. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

Note 1. Registers related to the ULPT: ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, and ULPTCMSR

**Condition C: In count start mode or count restart mode (ULPTMR3.TEECTL[1:0] bits = 10 or 11)**

When the counter is stopped (with the TCSTF bit of the ULPTCR register set to 0), even if 1 is set to the TSTART bit of the ULPTCR register and then the counter is started on an edge trigger of the ULPTEEn pin, the TCSTF bit remains at 0 (counter stopped) for five cycles of the count source.

Do not access the registers<sup>\*1</sup> related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 1 (counter running).

When the counter is running (with the TCSTF bit set to 1), even if 0 (stop the counter) is set to the TSTART bit, the TCSTF bit remains at 1 (counter running) for five cycles of the count source.

Do not access the registers<sup>\*1</sup> related to the ULPT other than the TCSTF bit until the TCSTF bit is set to 0 (counter stopped).

Before changing the TSTART bit from 0 to 1, clear the interrupt register. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

Note 1. Registers related to the ULPT: ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, and ULPTCMSR

### 23.5.2 Access to flags (TUNDF, TCMAF, and TCMBF bits in ULPTCR register)

It is recommended to set 1 to the TUNDF, TCMAF and TCMBF bits when changing the TSTART or TSTOP bits to prevent clearing TUNDF, TCMAF, or TCMBF by mistake. At this time, setting 1 to TUNDF, TCMAF and TCMBF bits is ignored.

### 23.5.3 Access to the ULPTCNT, ULPTCMA, and ULPTCMB registers

When setting to the same register sequentially while the counter is running (ULPTCR.TCSTF bit = 1), leave three or more cycles of count source between writes. The following registers are applicable:

- ULPTCNT
- ULPTCMA
- ULPTCMB

### 23.5.4 Mode changes

Registers related to the operating mode of the ULPT (ULPTMR1, ULPTMR2, ULPTMR3, ULPTIOC, ULPTISR, and ULPTCMSR) can be changed only while the counter is stopped (both the TSTART and TCSTF bits of the ULPTCR register are set to 0). Do not change these registers while the counter is running.

If a register related to the operating mode of the ULPT is changed, the values of the TUNDF, TCMAF, and TCMBF bits of the ULPTCR register are undefined. Before starting the counter, write 0 to the TUNDF, TCMAF, and TCMBF bits.

- TUNDF (no underflow)
- TCMAF (no match for compare match A)
- TCMBF (no match for compare match B)

### 23.5.5 Setting the ULPTOn, ULPTOAn, and ULPTOBn pins

To use the ULPTOn, ULPTOAn, and ULPTOBn pins as output pins, set up the ULPT, and then set PmnPFS.PMR bit to 1 after determining the initial output values.

### 23.5.6 Calculating the number of events

In event counter mode, the number of events is calculated by using the following formula:

Number of events = Initial counter value (ULPTCNT register write value) – Counter value at the end of a valid event

### 23.5.7 When the counter is stopped forcibly by the TSTOP bit

Do not access the ULPTCNT, ULPTCMA, ULPTCMB, ULPTCR, ULPTMR1, ULPTMR2, or ULPTMR3 register during one cycle of the count source after stopping the counter forcibly by the ULPTCR.TSTOP bit.

### 23.5.8 Digital filter

When using a digital filter, do not start the timer within five cycles of the digital filter clock after setting the ULPTIOC.TIPF[1:0] bits.

Also, when ULPTMR3.TEVPOL bit is changed while a digital filter is in use, do not start the timer within five cycles of the digital filter clock after setting the bit.

In Software Standby mode or Deep Software Standby mode1, the digital filter clock (PCLKB) stops and the digital filter does not function. Before entering Software standby mode or Deep Software Standby mode1, set the ULPTIOC.TIPF[1:0] bits to 00 (no filter).

### 23.5.9 Restrictions on Event Counting during Deep Software Standby Mode 1

If external events are to be counted continuously during Deep Software Standby mode 1, the ULPTEEn-DS and ULPTEVIn-DS pins should be held low during the transition to Deep Software Standby mode 1.

The device that generates the external events must be notified of the transition to Deep Software Standby mode 1 using the general I/O port.

An example of the procedure is shown below.(Examples of flow)

**(a) Procedure on the MCU**

1. During initial setup, set the port for notifying the external device as a general output port and output low.
2. Before transitioning to Deep Software Standby mode 1, set the PODR bit of the above port to 1.
3. Read the PODR bit and confirm that it is set to 1.
4. After 10  $\mu$ s has passed, execute the WFI instruction.

**(b) Procedure on the External Device**

1. When the above port goes high, hold the ULPTEEn-DS and ULPTEVIn-DS pins low within 10  $\mu$ s. If the time exceeds 10  $\mu$ s, add the excess time to the time in 4 above. If the time is less than 10  $\mu$ s, subtract the reduced time.
2. After 1 ms or more has passed, resume the event output.

### 23.5.10 Module-stop function

ULPT operation can be disabled or enabled using Module Stop Control Register E (MSTPCRE). The ULPT module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

### 23.5.11 Setting the ULPTEEn and ULPTEVIn pins

To use the ULPTEEn and ULPTEVIn pins as input pins, set up the ULPT, and then set PmnPFS.PMR bit to 1.

## 24. Realtime Clock (RTC)

### 24.1 Overview

The realtime clock (RTC) has two counting modes, calendar count mode and binary count mode, that are used by switching register settings. For calendar count mode, the RTC has a 100-year calendar from 2000 to 2099 and automatically adjusts dates for leap years. For binary count mode, the RTC counts seconds and retains the information as a serial value. Binary count mode can be used for calendars other than the Gregorian (Western) calendar.

The sub-clock oscillator or LOCO can be selected as the count source of the time counters. The RTC uses a 128-Hz clock acquired by dividing the count source by a prescaler. Year, month, date, day-of-week, a.m. /p.m. (in 12-hour mode), hour, minute, second, or 32-bit binary is counted by 1/128 second.

Table 24.1 lists the RTC specifications, Figure 24.1 shows a block diagram, and Table 24.2 lists the I/O pins.

**Table 24.1 RTC specifications**

Parameter	Specifications
Count mode	Calendar count mode/binary count mode
Count source*1	Sub-clock(XCIN), External clock input(EXCIN) or LOCO
Clock and calendar functions	<ul style="list-style-type: none"> <li>• Calendar count mode               <ul style="list-style-type: none"> <li>– Year, month, date, day of week, hour, minute, second are counted, BCD display</li> <li>– 12 hours/24 hours mode switching function</li> <li>– 30 seconds adjustment function (a number less than 30 is rounded down to 00 seconds, and 30 seconds or more are rounded up to 1 minute)</li> <li>– Automatic adjustment function for leap years</li> </ul> </li> <li>• Binary count mode               <ul style="list-style-type: none"> <li>– Count seconds in 32 bits, binary display</li> </ul> </li> <li>• Shared by both modes               <ul style="list-style-type: none"> <li>– Start/stop function</li> <li>– The sub-second digit is displayed in binary units (1 Hz, 2 Hz, 4 Hz, 8 Hz, 16 Hz, 32 Hz, or 64 Hz)</li> <li>– Clock error correction function</li> <li>– Clock (1-Hz/64-Hz) output</li> </ul> </li> </ul>
Interrupts	<ul style="list-style-type: none"> <li>• Alarm interrupt (RTC_ALM)               <ul style="list-style-type: none"> <li>– As an alarm interrupt condition, selectable for comparison with the following:                   <ul style="list-style-type: none"> <li>• Calendar count mode: Year, month, date, day-of-week, hour, minute, or second can be selected</li> <li>• Binary count mode: Each bit of the 32-bit binary counter</li> </ul> </li> </ul> </li> <li>• Periodic interrupt (RTC_PRD)               <ul style="list-style-type: none"> <li>– 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second can be selected as an interrupt period.</li> </ul> </li> <li>• Carry interrupt (RTC_CUP)               <ul style="list-style-type: none"> <li>– An interrupt is generated at either of the following conditions:                   <ul style="list-style-type: none"> <li>• When a carry from the 64-Hz counter to the second counter is generated.</li> <li>• When the 64-Hz counter is changed and the R64CNT register is read at the same time. (32-KHz count mode is only for 64-Hz counter reading)</li> </ul> </li> </ul> </li> <li>• Return from Software Standby or Deep Software Standby mode can be performed by the alarm interrupt or periodic interrupt</li> </ul>
Time capture function	<ul style="list-style-type: none"> <li>• Times can be captured when the edge of the time capture event input pin is detected. For every event input, month, date, hour, minute, and second are captured or the 32-bit binary counter value is captured.</li> <li>• Interrupt can be generated when the edge of the time capture event input is detected. The time capture event input pin and IRQ are shared.</li> </ul>
Event link function	Periodic event output (RTC_PRD)
TrustZone filter	Security and Privilege attribution can be set

Note 1. The frequency of the peripheral module clock (PCLKB)  $\geq$  the frequency of the count source should be satisfied.



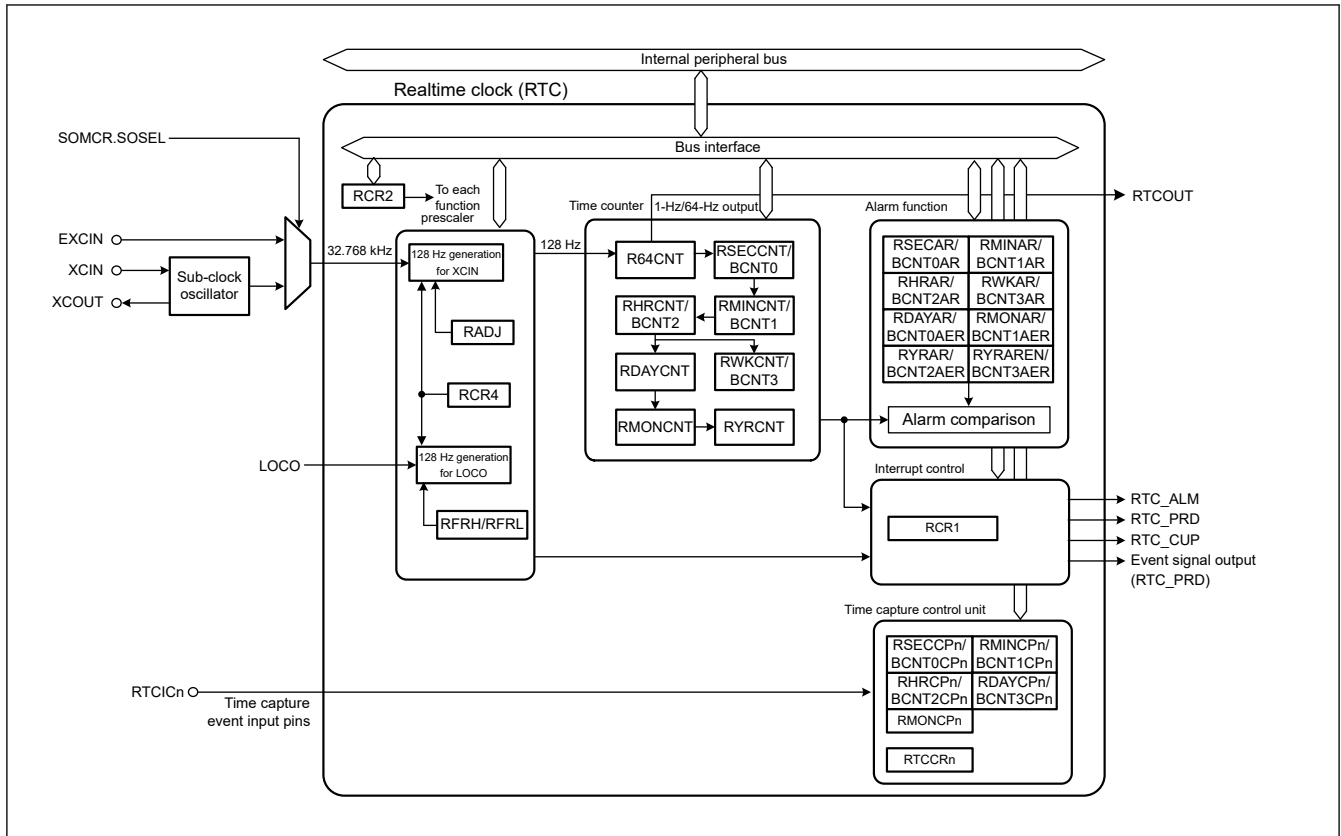


Figure 24.1 RTC block diagram

Table 24.2 RTC I/O pins

Pin name	I/O	Description
XCIN	Input	Connect a 32.768-kHz crystal to these pins
XCOU	Output	
EXCIN	Input	External sub-clock input
RTCOUT	Output	This pin is used to output a 1-Hz/64-Hz waveform, but not in Deep Software Standby mode
RTCICn (n = 0 to 2)	Input	Time capture event input pins RTCICn can be controlled by the VBTICTLR register. For more information, see <a href="#">section 11, Battery Backup Function</a> and <a href="#">section 19, I/O Ports</a> .

## 24.2 Register Descriptions

Write or read from the RTC registers as described in [section 24.6.5. Notes on Writing to and Reading from Registers](#).

If the value in an RTC register after a reset is given as x (undefined bits) in the list, it is not initialized by a reset. When RTC enters the reset state or a low power state during counting operations, for example, while the RCR2.START bit is 1, the year, month, day of the week, date, hours, minutes, seconds, and 64-Hz counters continue to operate.

Note: A reset generated while writing to a register might destroy the register value. In addition, do not allow the MCU to enter Software Standby mode, Deep Software Standby mode, or battery backup state immediately after setting any of these registers. For details, see [section 24.6.4. Transitions to Low Power Modes after Setting Registers](#).

### 24.2.1 R64CNT : 64-Hz Counter

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
Value after reset:	0	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
0	F64HZ	64-Hz Flag This bit indicates the 64-Hz state of the sub-second digit.	R
1	F32HZ	32-Hz Flag This bit indicates the 32-Hz state of the sub-second digit.	R
2	F16HZ	16-Hz Flag This bit indicates the 16-Hz state of the sub-second digit.	R
3	F8HZ	8-Hz Flag This bit indicates the 8-Hz state of the sub-second digit.	R
4	F4HZ	4-Hz Flag This bit indicates the 4-Hz state of the sub-second digit.	R
5	F2HZ	2-Hz Flag This bit indicates the 2-Hz state of the sub-second digit.	R
6	F1HZ	1-Hz Flag This bit indicates the 1-Hz state of the sub-second digit.	R
7	—	This bit is read as 0.	R

Note: S-TYPE3, P-TYPE3

The R64CNT counter is used in both calendar count mode and binary count mode. The 64-Hz counter (R64CNT) generates the period for a second by counting up periods of the 128-Hz clock. The state in the sub-second range can be confirmed by reading this counter.

This counter is set to 0x00 by an RTC software reset or an execution of a 30-second adjustment. To read this counter, follow the procedure in [section 24.3.5. Reading 64-Hz Counter and Time.](#)

### 24.2.2 RSECCNT : Second Counter (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	SEC1[3:0]	1-Second Count Counts from 0 to 9 every second. When a carry is generated, 1 is added to the tens place.	R/W
6:4	SEC10[2:0]	10-Second Count Counts from 0 to 5 for 60-second counting.	R/W
7	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RSECCNT counter sets and counts the BCD-coded second value. It counts the carries generated once per second in the 64-Hz counter.

The setting range is decimal 00 to 59. The RTC does not operate normally if any other value is set. Before writing to this register, you must stop the count operation using the START bit in RCR2.

To read this counter, follow the procedure in [section 24.3.5. Reading 64-Hz Counter and Time](#).

### 24.2.3 RMINCNT : Minute Counter (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—		MIN10[2:0]			MIN1[3:0]		
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MIN1[3:0]	1-Minute Count Counts from 0 to 9 every minute. When a carry is generated, 1 is added to the tens place.	R/W
6:4	MIN10[2:0]	10-Minute Count Counts from 0 to 5 for 60-minute counting.	R/W
7	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RMINCNT counter sets and counts the BCD-coded minute value. It counts the carries generated once every minute in the second counter.

A value from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in [section 24.3.5. Reading 64-Hz Counter and Time](#).

### 24.2.4 RHRCNT : Hour Counter (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PM	HR10[1:0]		HR1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	HR1[3:0]	1-Hour Count Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	R/W
5:4	HR10[1:0]	10-Hour Count Counts from 0 to 2 once per carry from the ones place.	R/W
6	PM	AM/PM select for time counter setting. 0: AM 1: PM	R/W
7	—	The read value is undefined. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RHRCNT counter sets and counts the BCD-coded hour value. It counts the carries generated once per hour in the minute counter. The specifiable time differs based on the setting in the hours mode bit (RCR2.HR24):

- When the RCR2.HR24 bit is 0 – from 00 to 11 (in BCD).
- When the RCR2.HR24 bit is 1 – from 00 to 23 (in BCD).

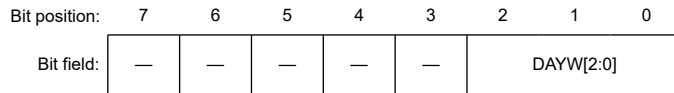
If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. The PM bit is only enabled when the RCR2.HR24 bit is 0.

Otherwise, the setting in the PM bit has no effect. To read this counter, follow the procedure in [section 24.3.5. Reading 64-Hz Counter and Time](#).

### 24.2.5 RWKCNT : Day-of-Week Counter (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x08



Value after reset: x x x x x x x x

Bit	Symbol	Function	R/W
2:0	DAYW[2:0]	Day-of-Week Counting 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting prohibited	R/W
7:3	—	The read values are undefined. The write value should be 0.	R/W

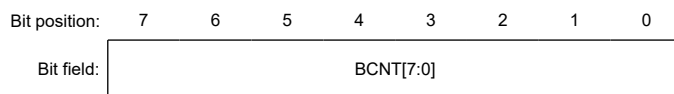
Note: S-TYPE3, P-TYPE3

The RWKCNT counter sets and counts in the coded day-of-week value. It counts the carries generated once per day in the hour counter. A value from 0 through 6 can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in [section 24.3.5. Reading 64-Hz Counter and Time](#).

### 24.2.6 BCNTn : Binary Counter n (n = 0 to 3) (in Binary Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x02 + 0x02 × n



Value after reset: x x x x x x x x

Bit	Symbol	Function	R/W
7:0	BCNT[7:0]	Binary Counter	R/W

Note: S-TYPE3, P-TYPE3

BCNTn is a read/write 8-bit register to access BCNT[31:0] that is a 32-bit binary counter. BCNT3 is assigned to the BCNT[31:24] bits, BCNT2 is assigned to the BCNT[23:16] bits, BCNT1 is assigned to the BCNT[15:8] bits, and BCNT0 is assigned to the BCNT[7:0] bits. BCNTn performs count operation by a carry generated for each second of the 64-Hz counter. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in [section 24.3.5. Reading 64-Hz Counter and Time](#).

### 24.2.7 RDAYCNT : Day Counter

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x0A

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	DATE10[1:0]	DATE1[3:0]
------------	---	---	-------------	------------

Value after reset: 0 0 x x x x x x

Bit	Symbol	Function	R/W
3:0	DATE1[3:0]	1-Day Count Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place.	R/W
5:4	DATE10[1:0]	10-Day Count Counts from 0 to 3 once per carry from the ones place.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RDAYCNT counter is used in calendar count mode to set and count the BCD-coded date value. It counts the carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year. Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.

A value from 01 through 31 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. When specifying a value, the range of specifiable days depends on the month and whether the year is a leap year. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in [section 24.3.5. Reading 64-Hz Counter and Time](#).

### 24.2.8 RMONCNT : Month Counter

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x0C

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	—	MON10	MON1[3:0]
------------	---	---	---	-------	-----------

Value after reset: 0 0 0 x x x x x

Bit	Symbol	Function	R/W
3:0	MON1[3:0]	1-Month Count Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
4	MON10	10-Month Count Counts from 0 to 1 once per carry from the ones place.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RMONCNT counter is used in calendar count mode to set and count the BCD-coded month value. It counts the carries generated once per month in the date counter.

A value from 01 through 12 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in [section 24.3.5. Reading 64-Hz Counter and Time](#).

### 24.2.9 RYRCNT : Year Counter

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x0E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	YR10[3:0]				YR1[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	YR1[3:0]	1-Year Count Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
7:4	YR10[3:0]	10-Year Count Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RYRCNT counter is used in calendar count mode to set and count the BCD-coded year value. It counts the carries generated once per year in the month counter.

A value from 00 through 99 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, you must stop the count operation using the START bit in RCR2. To read this counter, follow the procedure in [section 24.3.5. Reading 64-Hz Counter and Time](#).

### 24.2.10 RSECAR : Second Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENB	SEC10[2:0]			SEC1[3:0]			
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	SEC1[3:0]	1 Second Value for the ones place of seconds.	R/W
6:4	SEC10[2:0]	10 Seconds Value for the tens place of seconds.	R/W
7	ENB	ENB 0: Do not compare register value with RSECCNT counter value 1: Compare register value with RSECCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RSECAR is an alarm register associated with the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR

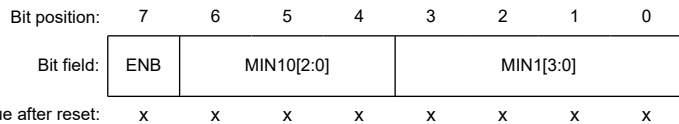
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC\_ALM interrupt is set to 1. RSECAR values from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

### 24.2.11 RMINAR : Minute Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
 RTC\_NS = 0x5020\_2000

Offset address: 0x12



Bit	Symbol	Function	R/W
3:0	MIN1[3:0]	1 Minute Value for the ones place of minutes.	R/W
6:4	MIN10[2:0]	10 Minutes Value for the tens place of minutes.	R/W
7	ENB	ENB 0: Do not compare register value with RMINCNT counter value 1: Compare register value with RMINCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RMINAR is an alarm register associated with the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

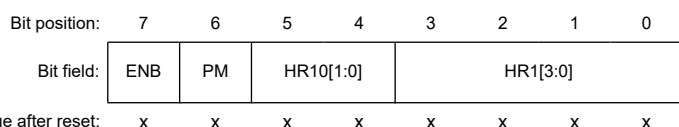
- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC\_ALM interrupt is set to 1. RMINAR values from 00 through 59 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

### 24.2.12 RHRAR : Hour Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
 RTC\_NS = 0x5020\_2000

Offset address: 0x14



Bit	Symbol	Function	R/W
3:0	HR1[3:0]	1 Hour Value for the ones place of hours.	R/W
5:4	HR10[1:0]	10 Hours Value for the tens place of hours.	R/W
6	PM	AM/PM select for alarm setting. 0: AM 1: PM	R/W
7	ENB	ENB 0: Do not compare register value with RHCNT counter value 1: Compare register value with RHCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RHRAR is an alarm register associated with the BCD-coded hour counter RHCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC\_ALM interrupt is set to 1. The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24):

- When the RCR2.HR24 bit is 0 – From 00 to 11 (in BCD).
- When the RCR2.HR24 bit is 1 – From 00 to 23 (in BCD).

If a value outside of this range is specified, the RTC does not operate correctly. When the RCR2.HR24 bit is 0, you must set the PM bit. When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect. This register is set to 0x00 by an RTC software reset.

### 24.2.13 RWKAR : Day-of-Week Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x16

Bit position: 7 6 5 4 3 2 1 0

Bit field:	ENB	—	—	—	—	DAYW[2:0]	
------------	-----	---	---	---	---	-----------	--

Value after reset: x x x x x x x x

Bit	Symbol	Function	R/W
2:0	DAYW[2:0]	Day-of-Week Setting 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting prohibited	R/W
6:3	—	The read values are undefined. The write value should be 0.	R/W



Bit	Symbol	Function	R/W
7	ENB	ENB 0: Do not compare register value with RWKCNT counter value 1: Compare register value with RWKCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RWKAR is an alarm register associated with the coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

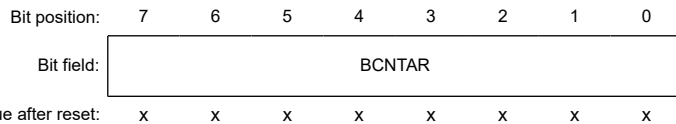
- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC\_ALM interrupt is set to 1. RWKAR values from 0 through 6 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

#### 24.2.14 BCNTnAR : Binary Counter n Alarm Register (n = 0 to 3) (in Binary Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x10 + 0x02 × n



Bit	Symbol	Function	R/W
7:0	BCNTAR	Alarm register associated with the 32-bit binary counter	R/W

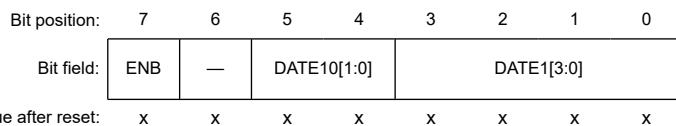
Note: S-TYPE3, P-TYPE3

BCNTnAR is a read/write alarm register associated with the 32-bit binary counter. BCNT3AR is assigned to the BCNTAR[31:24] bits, BCNT2AR is assigned to the BCNTAR[23:16] bits, BCNT1AR is assigned to the BCNTAR[15:8] bits, and BCNT0AR is assigned to the BCNTAR[7:0]. This register is set to 0x00 by an RTC software reset.

#### 24.2.15 RDAYAR : Date Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x18



Bit	Symbol	Function	R/W
3:0	DATE1[3:0]	1 Day Value for the ones place of days.	R/W

Bit	Symbol	Function	R/W
5:4	DATE10[1:0]	10 Days Value for the tens place of days.	R/W
6	—	The read value is undefined. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with RDAYCNT counter value 1: Compare register value with RDAYCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RDAYAR is an alarm register associated with the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

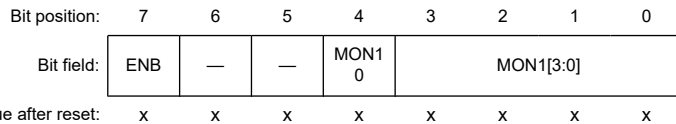
- RSECAR
- RMINAR
- RHRAR
- RWKAR
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC\_ALM interrupt is set to 1. The RDAYAR values from 01 through 31 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

### 24.2.16 RMONAR : Month Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x1A



Bit	Symbol	Function	R/W
3:0	MON1[3:0]	1 Month Value for the ones place of months.	R/W
4	MON10	10 Months Value for the tens place of months.	R/W
6:5	—	The read values are undefined. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with RMONCNT counter value 1: Compare register value with RMONCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

RMONAR is an alarm register associated with the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR

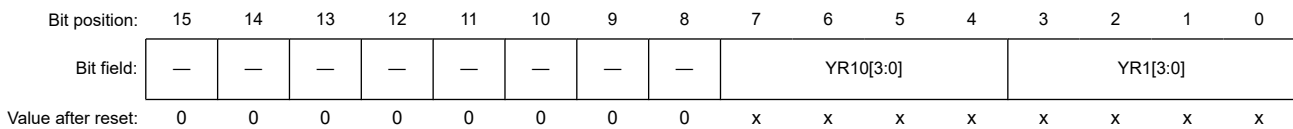
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC\_ALM interrupt is set to 1. The RMONAR values from 01 through 12 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x00 by an RTC software reset.

### 24.2.17 RYRAR : Year Alarm Register (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
 RTC\_NS = 0x5020\_2000

Offset address: 0x1C



Bit	Symbol	Function	R/W
3:0	YR1[3:0]	1 Year Value for the ones place of years.	R/W
7:4	YR10[3:0]	10 Years Value for the tens place of years.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

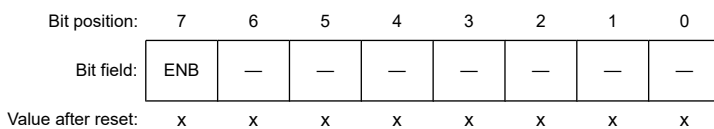
Note: S-TYPE3, P-TYPE3

RYRAR is an alarm register associated with the BCD-coded year counter RYRCNT. The RYRAR values from 00 through 99 (in BCD) can be specified. If a value outside of this range is specified, the RTC does not operate correctly. This register is set to 0x0000 by an RTC software reset.

### 24.2.18 RYRAREN : Year Alarm Enable Register (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
 RTC\_NS = 0x5020\_2000

Offset address: 0x1E



Bit	Symbol	Function	R/W
6:0	—	The read values are undefined. The write value should be 0.	R/W
7	ENB	ENB 0: Do not compare register value with the RYRCNT counter value 1: Compare register value with the RYRCNT counter value	R/W

Note: S-TYPE3, P-TYPE3

When the ENB bit in the RYRAREN register is set to 1, the RYRAR value is compared with the RYRCNT value. From the following alarm registers, only those selected with the ENB bits set to 1 are compared with the associated counters:

- RSECAR
- RMINAR
- RHRAR
- RWKAR

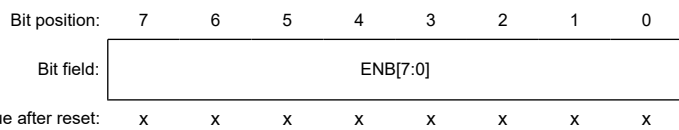
- RDAYAR
- RMONAR
- RYRAREN

When all the respective values match, the IR flag associated with the RTC\_ALM interrupt is set to 1. This register is set to 0x00 by an RTC software reset.

### 24.2.19 BCNTnAER : Binary Counter n Alarm Enable Register (n = 0 to 3) (in Binary Count Mode)

Base address: RTC = 0x4020\_2000  
 RTC\_NS = 0x5020\_2000

Offset address: 0x18 + 0x02 × n



Bit	Symbol	Function	R/W
7:0	ENB[7:0]	Setting the alarm enable associated with the 32-bit binary counter	R/W

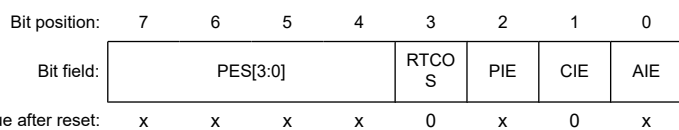
Note: S-TYPE3, P-TYPE3

BCNTnAER is a read/write register for setting the alarm enable (BCNTAER) associated with the 32-bit binary counter. BCNT3AER is assigned to the BCNTAER.ENB[31:24] bits, BCNT2AER register is assigned to the BCNTAER.ENB[23:16] bits, BCNT1AER is assigned to the BCNTAER.ENB[15:8] bits, and BCNT0AER is assigned to the BCNTAER.ENB[7:0] bits. The binary counter (BCNT[31:0]) associated with the BCNTAER.ENB[31:0] bits that are set to 1 is compared with the binary alarm register (BCNTAR) and, when all match, the IR flag associated with the RTC\_ALM interrupt is set to 1. This register is set to 0x00 by an RTC software reset.

### 24.2.20 RCR1 : RTC Control Register 1

Base address: RTC = 0x4020\_2000  
 RTC\_NS = 0x5020\_2000

Offset address: 0x22



Bit	Symbol	Function	R/W
0	AIE	Alarm Interrupt Enable 0: Disable alarm interrupt requests 1: Enable alarm interrupt requests	R/W
1	CIE	Carry Interrupt Enable 0: Disable carry interrupt requests 1: Enable carry interrupt requests	R/W
2	PIE	Periodic Interrupt Enable 0: Disable periodic interrupt requests 1: Enable periodic interrupt requests	R/W
3	RTCOS	RTCOUT Output Select 0: Outputs 1 Hz on RTCOUT 1: Outputs 64 Hz RTCOUT	R/W

Bit	Symbol	Function	R/W
7:4	PES[3:0]	Periodic Interrupt Select 0x6: Generate periodic interrupt every 1/256 second*1 0x7: Generate periodic interrupt every 1/128 second 0x8: Generate periodic interrupt every 1/64 second 0x9: Generate periodic interrupt every 1/32 second 0xA: Generate periodic interrupt every 1/16 second 0xB: Generate periodic interrupt every 1/8 second 0xC: Generate periodic interrupt every 1/4 second 0xD: Generate periodic interrupt every 1/2 second 0xE: Generate periodic interrupt every 1 second 0xF: Generate periodic interrupt every 2 seconds Others: Do not generate periodic interrupts	R/W

Note: S-TYPE3, P-TYPE3

Note 1. When LOCO is selected (RCR4.RCKSEL = 1) while PES[3:0] = 0x6, a periodic interrupt is generated every 1/128 second.

The RCR1 register is used in both calendar count mode and binary count mode. Bits AIE, PIE, and PES[3:0] are updated synchronously with the count source. When the RCR1 register is modified, check that all the bits are updated before proceeding.

### AIE bit (Alarm Interrupt Enable)

The AIE bit enables or disables alarm interrupt requests.

If the times indicated in the counters and alarm settings match in Deep Software Standby mode, the MCU returns from Deep Software Standby mode regardless of the AIE bit value.

### CIE bit (Carry Interrupt Enable)

The CIE bit enables or disables interrupt requests when a carry to the RSECCNT/BCNT0 register occurs, or when a carry to the 64-Hz counter (R64CNT) occurs while reading the 64-Hz counter.

### PIE bit (Periodic Interrupt Enable)

The PIE bit enables or disabled a periodic interrupt.

If the periods indicated in the counters and PES[3:0] settings match in Deep Software Standby mode, the MCU returns from Deep Software Standby mode regardless of the PIE bit value.

### RTCOS bit (RTCOU Output Select)

The RTCOS bit selects the RTCOUT output period. The RTCOS bit must be rewritten while the count operation is stopped (RCR2.START = 0) and the RTCOUT output is disabled (RCR2.RTCOE = 0). When RTCOUT is output to an external pin, the RCR2.RTCOE bit must be enabled.

### PES[3:0] bits (Periodic Interrupt Select)

The PES[3:0] bits specify the period for the periodic interrupt. A periodic interrupt is generated with the period specified in these bits.

#### 24.2.21 RCR2 : RTC Control Register 2 (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CNTM D	HR24	AADJ P	AADJ E	RTCO E	ADJ30	RESE T	START

Value after reset: x x x x 0 0 0 x

Bit	Symbol	Function	R/W
0	START	Start 0: Stop prescaler and time counter 1: Operate prescaler and time counter normally	R/W

Bit	Symbol	Function	R/W
1	RESET	RTC Software Reset 0: In writing: Invalid (writing 0 has no effect). In reading: Normal time operation in progress, or an RTC software reset has completed. 1: In writing: Initialize the prescaler and target registers for RTC software reset*1. In reading: RTC software reset in progress.	R/W
2	ADJ30	30-Second Adjustment 0: In writing: Invalid (writing 0 has no effect). In reading: Normal time operation in progress, or 30-second adjustment has completed. 1: In writing: Execute 30-second adjustment. In reading: 30-second adjustment in progress.	R/W
3	RTCOE	RTCOUT Output Enable 0: Disable RTCOUT output 1: Enable RTCOUT output	R/W
4	AADJE	Automatic Adjustment Enable*2*3 0: Disable automatic adjustment 1: Enable automatic adjustment	R/W
5	AADJP	Automatic Adjustment Period Select*2*3 0: The RADJ.ADJ[5:0] setting from the count value of the prescaler every minute. 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every 10 seconds.	R/W
6	HR24	Hours Mode*3 0: Operate RTC in 12-hour mode 1: Operate RTC in 24-hour mode	R/W
7	CNTMD	Count Mode Select*4 0: Calendar count mode 1: Binary count mode	R/W

Note: S-TYPE3, P-TYPE3

Note 1. R64CNT, RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, RYRAR, RYRAREN, RADJ, RTCCRN, RSECCPN, RMINCPN, RHRCPN, RDAYCPN, RMONCPN, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP.

Note 2. When LOCO is selected, the setting of this bit is disabled.

Note 3. When rewriting this bit, confirm that the value has been rewritten before performing the following processing. See [section 24.6.5. Notes on Writing to and Reading from Registers](#) for notes on register writing/reading.

Note 4. When rewriting this bit, confirm that the value has been rewritten before performing the following processing.

The RCR2 register is related to hours mode, automatic adjustment function, enabling RTCOUT output, 30-second adjustment, RTC software reset, and controlling count operation.

### START bit (Start)

The START bit stops or restarts the prescaler or time counter operation. This bit is updated in synchronization with the next cycle of the count source. When the START bit is modified, check that the bit is updated before proceeding.

### RESET bit (RTC Software Reset)

The RESET bit initializes the prescaler and registers to be reset by RTC software. When 1 is written to this bit, initialization starts in synchronization with the count source. When the initialization is complete, the RESET bit is automatically set to 0. Check that this bit is 0 before proceeding.

### ADJ30 bit (30-Second Adjustment)

The ADJ30 bit is for 30-second adjustment.

When 1 is written to the ADJ30 bit, the RSECCNT value of less than 30 seconds is rounded down to 00 second and the value of 30 seconds or more is rounded up to 1 minute.

The 30-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ30 bit is automatically set to 0 after the 30-second adjustment completes. If 1 is written to the ADJ30 bit, check that the bit is 0 before proceeding. When the 30-second adjustment is performed, the prescaler and R64CNT are also reset. The ADJ30 bit is set to 0 by an RTC software reset.

**RTCOE bit (RTCOUT Output Enable)**

The RTCOE bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.

Use the START bit to stop counting before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.

When RTCOUT is to be output from an external pin, enable the RTCOE bit and set up the port control for the pin.

**AADJE bit (Automatic Adjustment Enable)**

The AADJE bit controls (enables or disables) automatic adjustment.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit.

The AADJE bit is set to 0 by an RTC software reset.

**AADJP bit (Automatic Adjustment Period Select)**

The AADJP bit selects the automatic-adjustment period.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit.

The AADJP bit is set to 0 by an RTC software reset.

**HR24 bit (Hours Mode)**

The HR24 bit specifies whether the RTC operates in 12- or 24-hour mode.

Use the START bit to stop counting before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.

**CNTMD bit (Count Mode Select)**

The CNTMD bit specifies whether the RTC count mode operates in calendar count mode or in binary count mode.

When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated in synchronization with the count source. However, the count mode switches only after the RTC software reset. (Bit switches before RTC reset, mode switches after RTC reset.)

For details on initial settings, see [section 24.3.1. Outline of Initial Settings of Registers after Power On](#).

**24.2.22 RCR2 : RTC Control Register 2 (in Binary Count Mode)**

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x24

Bit position: 7 6 5 4 3 2 1 0

Bit field:	CNTM D	—	AADJ P	AADJ E	RTCO E	—	RESE T	START
------------	-----------	---	-----------	-----------	-----------	---	-----------	-------

Value after reset: x x x x 0 0 0 x

Bit	Symbol	Function	R/W
0	START	Start 0: Stop the 32-bit binary counter, 64-Hz counter, and prescaler 1: Operate the 32-bit binary counter, 64-Hz counter, and prescaler normally	R/W
1	RESET	RTC Software Reset 0: In writing: Invalid (writing 0 has no effect). In reading: Normal time operation in progress, or an RTC software reset has completed. 1: In writing: Initialize the prescaler and target registers for RTC software reset* <sup>1</sup> . In reading: RTC software reset in progress.	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
3	RTCOE	RTCOUT Output Enable 0: Disable RTCOUT output 1: Enable RTCOUT output	R/W
4	AADJE	Automatic Adjustment Enable <sup>*2*3</sup> 0: Disable automatic adjustment 1: Enable automatic adjustment	R/W
5	AADJP	Automatic Adjustment Period Select <sup>*2*3</sup> 0: Add or subtract RADJ.ADJ [5:0] bits from prescaler count value every 32 seconds 1: Add or subtract RADJ.ADJ [5:0] bits from prescaler count value every 8 seconds.	R/W
6	—	The read value is undefined. The write value should be 0.	R/W
7	CNTMD	Count Mode Select <sup>*4</sup> 0: Calendar count mode 1: Binary count mode	R/W

Note: S-TYPE3, P-TYPE3

Note 1. R64CNT, BCNTnAR, BCNTnAER, RADJ, RTCCRN, BCNTnCPm, RCR2.ADJ30, RCR2.AADJE, RCR2.AADJP.

Note 2. When LOCO is selected, the setting of this bit is disabled.

Note 3. When rewriting this bit, confirm that the value has been rewritten before performing the following processing. See [section 24.6.5. Notes on Writing to and Reading from Registers](#) for notes on register writing/reading.

Note 4. When rewriting this bit, confirm that the value has been rewritten before performing the following processing.

RCR2 in the binary count mode is a register related to the automatic correction function, RTCOUT output enable, RTC software reset, and count mode control.

### START bit (Start)

The START bit stops or restarts the prescaler or counter (clock) operation. This bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated before proceeding.

### RESET bit (RTC Software Reset)

The RESET bit initializes the prescaler and registers to be reset by RTC software. When 1 is written to this bit, initialization starts in synchronization with the count source. When the initialization is complete, the RESET bit is automatically set to 0. When 1 is written to the RESET bit, check that the bit is 0 before proceeding.

### RTCOE bit (RTCOUT Output Enable)

The RTCOE bit enables output of a 1-Hz/64-Hz clock signal from the RTCOUT pin.

Use the START bit to stop counting before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time. When an RTCOUT signal is to be output from an external pin, enable the port control in addition to setting this bit.

### AADJE bit (Automatic Adjustment Enable)

The AADJE bit controls (enables or disables) automatic adjustment.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJE bit. The AADJE bit is set to 0 by an RTC software reset.

### AADJP bit (Automatic Adjustment Period Select)

The AADJP bit selects the automatic-adjustment period.

Correction period can be selected from 32 second units or 8 second units in binary count mode.

Set the plus-minus bits (RADJ.PMADJ[1:0]) to 00b (adjustment is not performed) before changing the value of the AADJP bit. The AADJP bit is set to 0 by an RTC software reset.

### CNTMD bit (Count Mode Select)

The CNTMD bit specifies whether the RTC count mode operates in calendar count mode or in binary count mode.

When setting the count mode, execute an RTC software reset and start again from the initial settings. This bit is updated in synchronization with the count source. However, the count mode switches only after the RTC software reset. (Bit switches before RTC reset, mode switches after RTC reset.)



For details on initial settings, see [section 24.3.1. Outline of Initial Settings of Registers after Power On](#).

### 24.2.23 RCR4 : RTC Control Register 4

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x28

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RCKSEL
Value after reset:	0	0	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	RCKSEL	Count Source Select 0: Sub-clock oscillator is selected 1: LOCO is selected	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RCR4 register is used in both calendar count mode and binary count mode.

#### RCKSEL bit (Count Source Select)

The RCKSEL bit selects the count source from the sub-clock oscillator and LOCO.

The RCKSEL bit is only used in normal operation mode. When the RCKSEL bit is set to 0, the time is counted with the sub-clock oscillator. When the bit is set to 1, the time is counted with LOCO.

For details on count source setting, see [section 24.3.1. Outline of Initial Settings of Registers after Power On](#) and [section 24.3.2. Clock and Count Mode Setting Procedure](#). The count source must be selected only once before specifying the initial settings of the RTC registers at power on.

### 24.2.24 RFRL : Frequency Register L

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x2C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFC[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
15:0	RFC[15:0]	Frequency Comparison Value Write 0x00FF to this register when using the LOCO.	R/W

Note: S-TYPE3, P-TYPE3

RFRL is a register for controlling the prescaler when LOCO is selected.

The RTC time counter operates on a 128-Hz clock signal as the base clock. Therefore, when LOCO is selected, LOCO is divided by the prescaler to generate a 128-Hz clock signal. Set the frequency comparison value in the RFC[15:0] bits to generate a 128-Hz clock from the LOCO frequency. Before writing to RFC[15:0] after a cold start, write 0x0000 to the RFRH register.

A value from 0x0007 through 0x01FF can be specified as the frequency comparison value. If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2. The operating frequency of the peripheral module clock and the LOCO should be such that the peripheral module clock is  $\geq$  LOCO.

Calculation method of frequency comparison value:

$$RFC[15:0] = (\text{LOCO clock frequency}) / 128 - 1$$

When the LOCO frequency is 32.768 kHz, the RFRL register should be set to 0x00FF.

### 24.2.25 RFRH : Frequency Register H

Base address: RTC = 0x4020\_2000  
 RTC\_NS = 0x5020\_2000

Offset address: 0x2A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFC16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	RFC16	Write 0 before writing to the RFRL register after a cold start.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Before writing to RFRHL.RFC[15:0] after a cold start, write 0x0000 to the RFRH register.

### 24.2.26 RADJ : Time Error Adjustment Register

Base address: RTC = 0x4020\_2000  
 RTC\_NS = 0x5020\_2000

Offset address: 0x2E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PMADJ[1:0]		ADJ[5:0]					
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
5:0	ADJ[5:0]	Adjustment Value These bits specify the adjustment value from the prescaler.	R/W
7:6	PMADJ[1:0]	Plus-Minus 0 0: Do not perform adjustment. 0 1: Adjustment is performed by the addition to the prescaler 1 0: Adjustment is performed by the subtraction from the prescaler 1 1: Setting prohibited.	R/W

Note: S-TYPE3, P-TYPE3

The RADJ register is used in both calendar count mode and binary count mode. Adjustment is performed by the addition to or subtraction from the prescaler or 64-Hz counter. If the Automatic Adjustment Enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ. If the RCR2.AADJE bit is 1, adjustment is performed in the interval specified in the Automatic Adjustment Period Select (RCR2.AADJP) bit.

The current adjustment by software (disabling automatic adjustment) may be invalid if the following adjustment value is specified within 320 cycles of the count source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the count source after the register setting, then specify the next adjustment value.

RADJ is updated in synchronization with the count source. When RADJ is modified, check that all the bits are updated before continuing with more processing. This register is set to 0x00 by an RTC software reset. The setting of this register is enabled only when the sub-clock oscillator is selected. When LOCO is selected, adjustment is not performed.

#### ADJ[5:0] bits (Adjustment Value)

The ADJ[5:0] bits specify the adjustment value (number of sub-clock cycles) from the prescaler.

**PMADJ[1:0] bits (Plus-Minus)**

The PMADJ[1:0] bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ[5:0] bits.

**24.2.27 RTCCRn : Time Capture Control Register n (n = 0 to 2)**

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x40 + 0x02 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCEN	—	TCNF[1:0]	—	TCST	TCCT[1:0]		

Value after reset: x 0 x x 0 x x x

Bit	Symbol	Function	R/W
1:0	TCCT[1:0]	Time Capture Control 0 0: Do not detect events 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
2	TCST	Time Capture Status 0: No event detected 1: Event detected <sup>*1</sup>	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	TCNF[1:0]	Time Capture Noise Filter Control 0 0: Turn noise filter off 0 1: Setting prohibited 1 0: Turn noise filter on (count source) 1 1: Turn noise filter on (count source by divided by 32)	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TCEN	Time Capture Event Input Pin Enable 0: Disable the RTCICn pin as the time capture event input pin 1: Enable the RTCICn pin as the time capture event input pin	R/W

Note: S-TYPE3, P-TYPE3

Note 1. Indicates that an event is detected. Writing 1 to this bit has no effect. Writing 0 sets this bit to 0.

The RTCCRn register is used both in calendar count mode and in binary count mode. RTCCR0, RTCCR1, and RTCCR2 control the RTCIC0, RTCIC1, and RTCIC2 pins, respectively.

RTCCRn is updated in synchronization with the count source. When RTCCRn is modified, check that all the bits except the TCST bit are updated before continuing with additional processing. This register is cleared to 0x00 by an RTC software reset. When RTCICm is used as the time capture pin, VBTICTLR.VCHnIEN (n = 0 to 2) must be set to 1.

**TCCT[1:0] bits (Time Capture Control)**

The TCCT[1:0] bits control the edge detection of the time capture event input pins, RTCIC0, RTCIC1, and RTCIC2. The detection edge is selectable. The TCCT[1:0] bits must be set while the VBTICTLR.VCHnIEN bit is 1.

**TCST bit (Time Capture Status)**

The TCST bit indicates that an event on the time capture event input pins, RTCIC0, RTCIC1, and RTCIC2, was detected. When the TCST bit is 0, no event is detected. When the TCST bit is 1, this bit indicates that an event was detected on the associated pin and the capture register is valid. When multiple events are detected, the capture time for the first event is retained.

The event is detected only during count operation (RCR2.START bit = 1). Before reading the capture register, make sure that this bit is set to 1.

Set the TCST bit while the TCCT[1:0] bits are 00b (no event is detected). The TCST bit is set to 0 in synchronization with the count source. When the TCST bit is set to 0, check that the bit is updated before continuing with additional processing.

**TCNF[1:0] bits (Time Capture Noise Filter Control)**

The TCNF[1:0] bits control the noise filter of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).

When the noise filter is on, the count source divided by 1 or divided by 32 is selectable. In this case, when the input level on the time capture event input pin matches three consecutive times at the set sampling period, the input level is determined.

Set the TCNF[1:0] bits while the TCCT[1:0] bits are 00b (no event is detected). When the noise filter is used, set the TCNF[1:0] bits, wait for 3 cycles of the specified sampling period, then set the TCCT[1:0] bits. Set the TCNF[1:0] bits when the VBTICTLR.VCHnIEN bit is 1.

**TCEN bit (Time Capture Event Input Pin Enable)**

The TCEN bit enables or disables the time capture event input pins RTCIC0, RTCIC1, and RTCIC2. When the functions of the time capture event input pins are multiplexed, set VBTICTLR first. If the TCEN bit is set to 0, also set the TCCT[1:0] bits to 00b.

Before setting this bit to 1, be sure to set the count source setting bit (RCR4.RCKSEL), RTC time capture event enable bit (RCPE.RTCEN), port control setting bits (PmnPFS.PDR, and PmnPFS.PMR). For details on the port control setting bits (PmnPFS.PDR and PmnPFS.PMR), see [section 19, I/O Ports](#).

**24.2.28 RSECCPn : Second Capture Register n (n = 0 to 2) (in Calendar Count Mode)**

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x52 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—		SEC10[2:0]			SEC1[3:0]		
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	SEC1[3:0]	1-Second Capture Capture value for the ones place of seconds.	R
6:4	SEC10[2:0]	10-Second Capture Capture value for the tens place of seconds.	R
7	—	The read value is undefined.	R

Note: S-TYPE3, P-TYPE3

RSECCPn is a read-only register that captures the RSECCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RSECCP0, RSECCP1, and RSECCP2 registers, respectively. This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

**24.2.29 RMINCPn : Minute Capture Register n (n = 0 to 2) (in Calendar Count Mode)**

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x54 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—		MIN10[2:0]			MIN1[3:0]		
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MIN1[3:0]	1-Minute Capture Capture value for the ones place of minutes.	R

Bit	Symbol	Function	R/W
6:4	MIN10[2:0]	10-Minute Capture Capture value for the tens place of minutes.	R
7	—	The read value is undefined.	R

Note: S-TYPE3, P-TYPE3

RMINCPn is a read-only register that captures the RMINCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMINCP0, RMINCP1, and RMINCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

### 24.2.30 RHRCPn : Hour Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x56 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PM	HR10[1:0]	HR1[3:0]				
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	HR1[3:0]	1-Hour Capture Capture value for the ones place of hours	R
5:4	HR10[1:0]	10-Hour Capture Capture value for the tens place of hours	R
6	PM	PM 0: AM 1: PM	R
7	—	The read value is undefined.	R

Note: S-TYPE3, P-TYPE3

RHRCPn is a read-only register that captures the RHRCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RHRCP0, RHRCP1, and RHRCP2 registers, respectively.

The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, you must stop the time capture event detection using the RTCCRn.TCCT[1:0] bits.

### 24.2.31 RDAYCPn : Date Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x5A + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	DATE10[1:0]	DATE1[3:0]				
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	DATE1[3:0]	1-Day Capture Capture value for the ones place of days.	R

Bit	Symbol	Function	R/W
5:4	DATE10[1:0]	10-Day Capture Capture value for the tens place of days.	R
7:6	—	The read value is undefined.	R

Note: S-TYPE3, P-TYPE3

RDAYCPn is a read-only register that captures the RDAYCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RDAYCP0, RDAYCP1, and RDAYCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

### 24.2.32 RMONCPn : Month Capture Register n (n = 0 to 2) (in Calendar Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x5C + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MON1 0	MON1[3:0]			
Value after reset:	0	0	0	x	x	x	x	x

Bit	Symbol	Function	R/W
3:0	MON1[3:0]	1-Month Capture Capture value for the ones place of months.	R
4	MON10	10-Month Capture Capture value for the tens place of months.	R
7:5	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

RMONCPn is a read-only register that captures the RMONCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMONCP0, RMONCP1, and RMONCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, the time capture event detection should be stopped using the RTCCRn.TCCT[1:0] bits.

### 24.2.33 BCNTnCPm : BCNTn Capture Register m (n = 0 to 3, m = 0 to 2) (in Binary Count Mode)

Base address: RTC = 0x4020\_2000  
RTC\_NS = 0x5020\_2000

Offset address: 0x52 + 0x10 × m (BCNT0CPm)  
0x54 + 0x10 × m (BCNT1CPm)  
0x56 + 0x10 × m (BCNT2CPm)  
0x5A + 0x10 × m (BCNT3CPm)

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	x	x	x	x	x	x	x	x

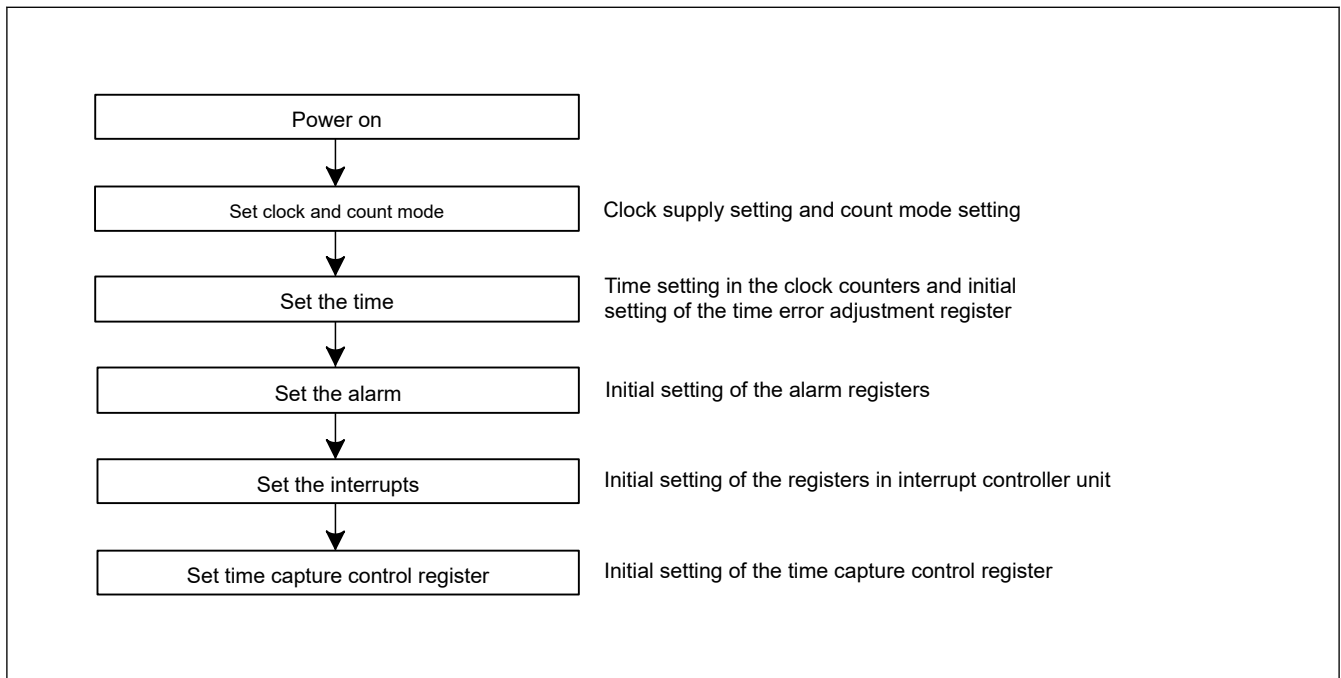
BCNTnCPm is a read-only register that captures the BCNTn value when a time capture event is detected. BCNT3CPm is assigned to the BCNTCPm[31:24] bits, BCNT2CPm is assigned to the BCNTCPm[23:16] bits, BCNT1CPm is assigned to the BCNTCPm[15:8] bits and BCNT0CPm is assigned to the BCNTCPm[7:0] bits. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the BCNTnCP0, BCNTnCP1, and BCNTnCP2 registers, respectively.

This register is cleared to 0x00 by an RTC software reset. Before reading from this register, you must stop the time capture event detection using the RTCCRn.TCCT[1:0] bits.

## 24.3 Operation

### 24.3.1 Outline of Initial Settings of Registers after Power On

After the power is turned on, perform the initial settings for the clock, count mode, time error adjustment, time, alarm, interrupts, and time capture.



**Figure 24.2** Outline of initial settings after a power on

### 24.3.2 Clock and Count Mode Setting Procedure

[Figure 24.3](#) shows how to set the clock and the count mode.

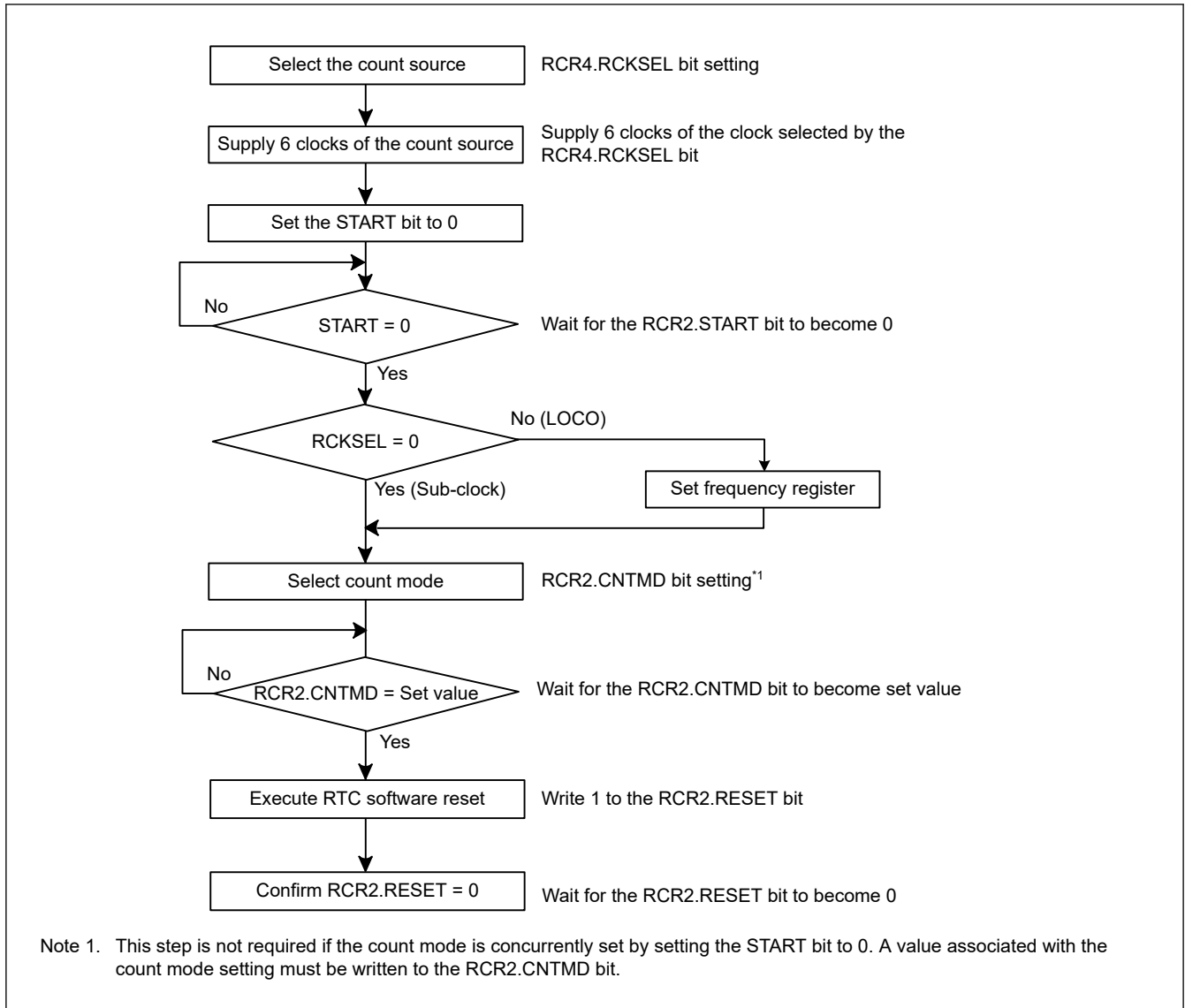
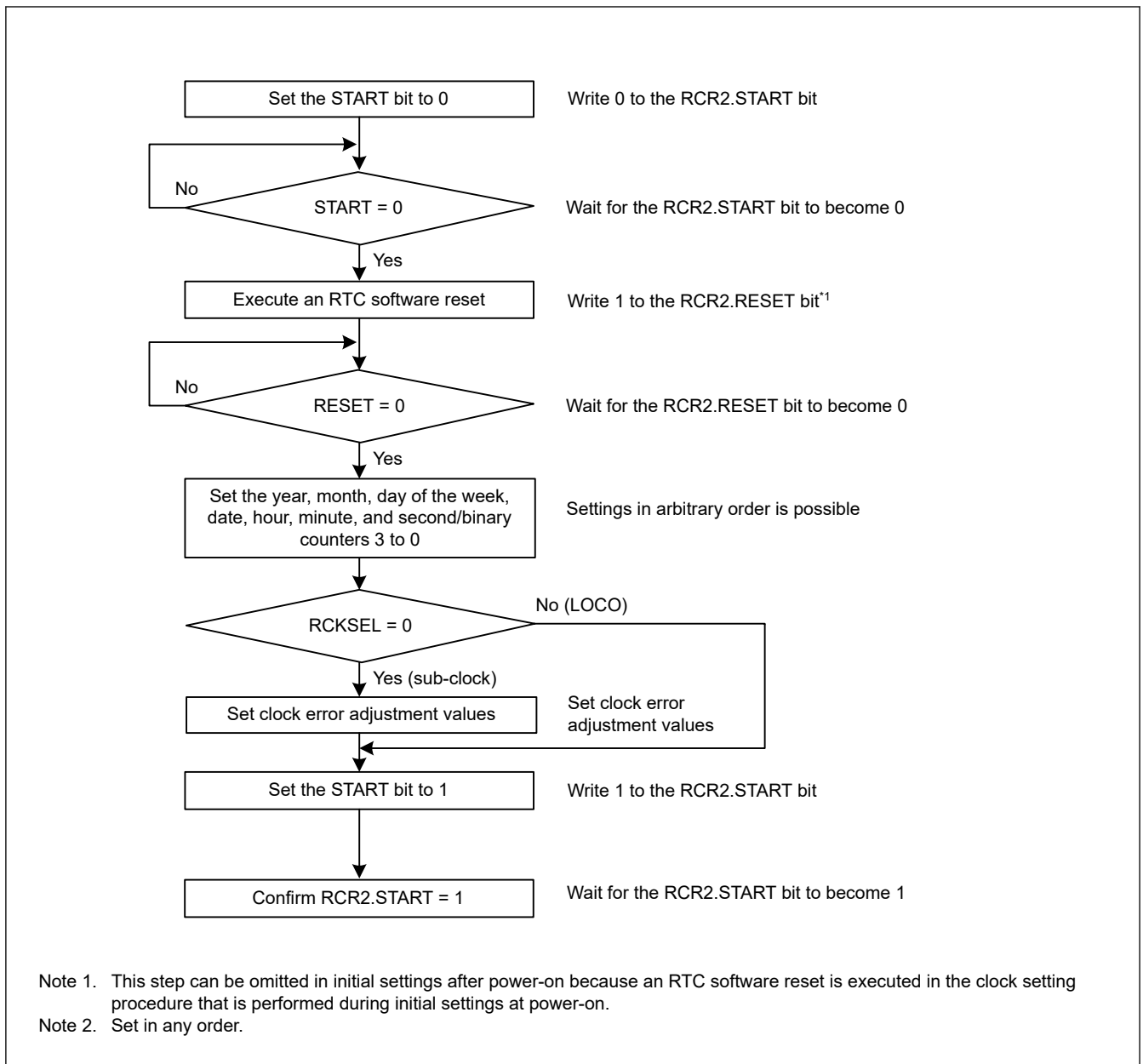


Figure 24.3 Clock and count mode setting procedure

### 24.3.3 Setting the Time

Figure 24.4 shows how to set the time.

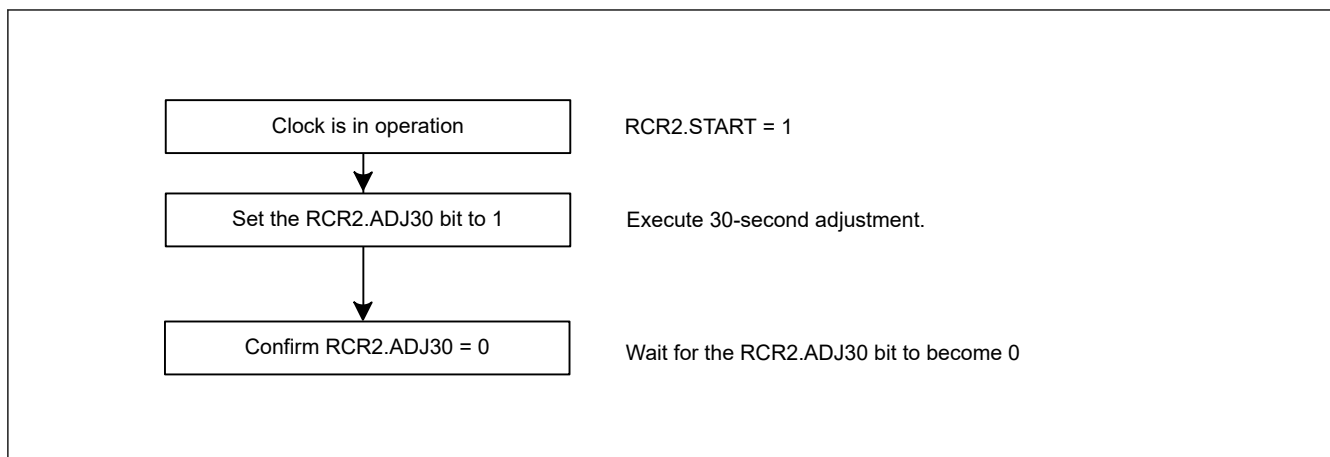




**Figure 24.4** Setting the time

### 24.3.4 30-Second Adjustment

Figure 24.5 shows how to execute a 30-second adjustment.



**Figure 24.5 30-second adjustment**

### 24.3.5 Reading 64-Hz Counter and Time

Figure 24.6 shows how to read a 64-Hz counter and time.

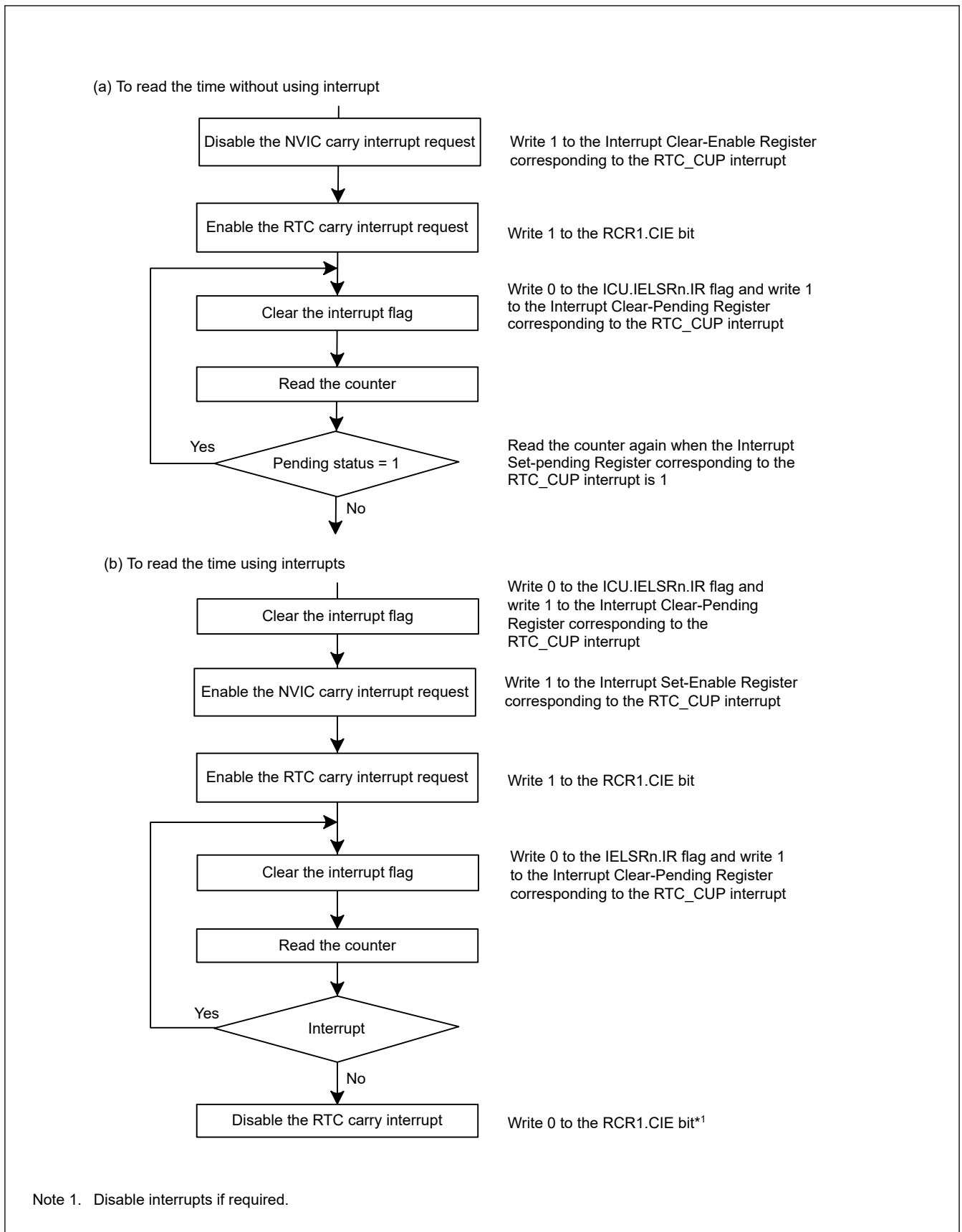
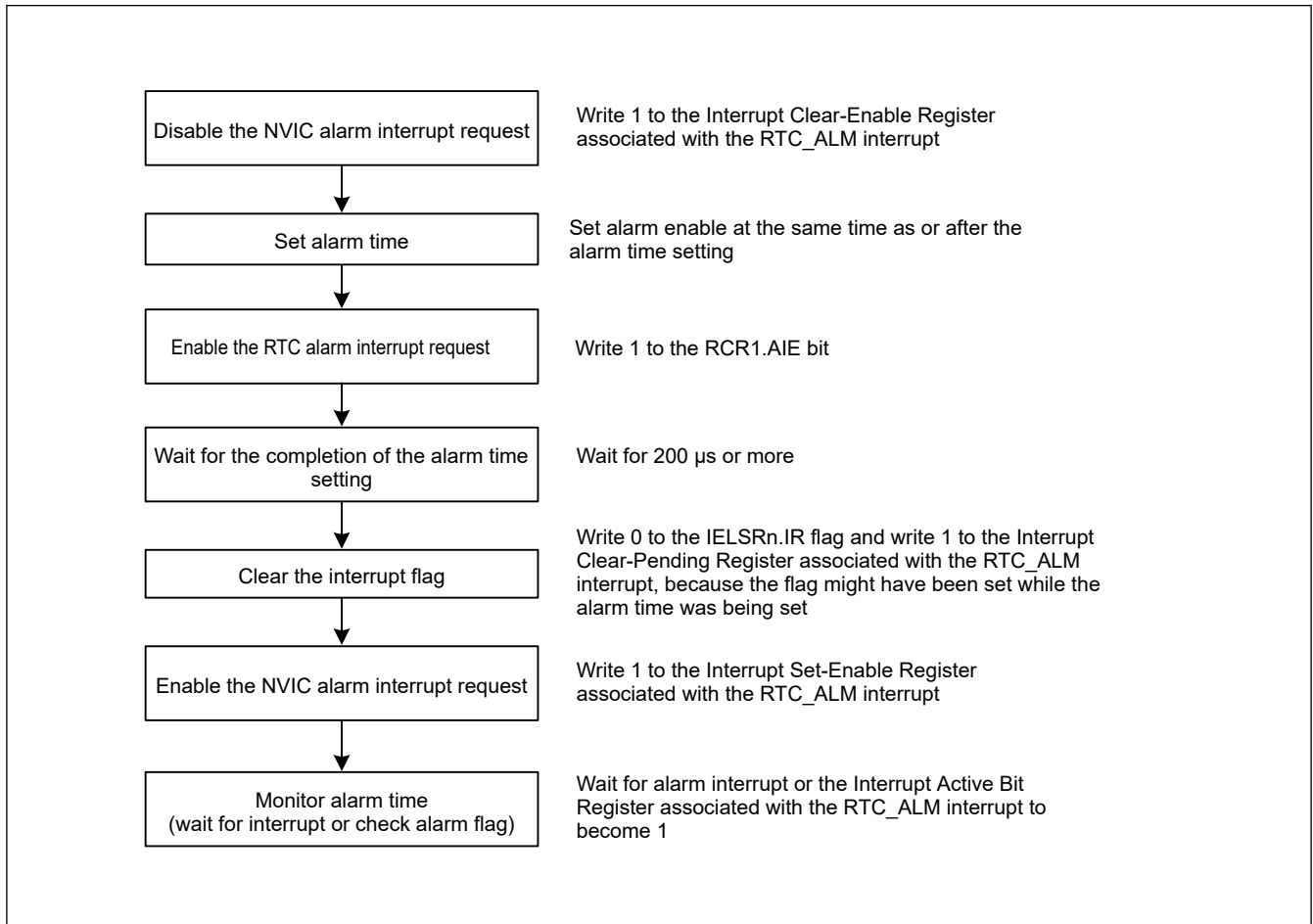


Figure 24.6 Reading time

If a carry occurs while the 64-Hz counter and time are read, the correct time is not obtained, therefore they must be read again. The procedure for reading the time without using interrupts is shown in (a) in Figure 24.6, and the procedure using carry interrupts is shown in (b). To keep the program simple, method (a) should be used in most cases.

### 24.3.6 Alarm Function

Figure 24.7 shows how to use the alarm function.



**Figure 24.7 Using the alarm function**

In calendar count mode, an alarm can be generated by any one of year, month, date, day-of-week, hour, minute or second, or any combination of those. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

In binary count mode, an alarm can be generated in any bit combination of 32 bits. Write 1 to the ENB bit of the Alarm Enable register associated with the target bit of the alarm, and set the alarm time in the alarm register. For bits that are not the target of the alarm, write 0 to the ENB bit of the Alarm Enable register.\*1

For any of the ENB[31:0] bits that are set to 1, the bits in the corresponding positions in the binary counter (BCNT[31:0]) are compared with the values of the corresponding bits in the binary alarm registers\*1. When all such bits match, the IR flag associated with the RTC\_ALM interrupt is set to 1 and the corresponding bits in the Interrupt Set-Pending/Clear-Pending Registers are set to 1. Alarm detection can be confirmed by reading the Interrupt Set-Pending Register associated with the RTC\_ALM interrupt, but an interrupt should be used in most cases. If 1 is set in the Interrupt Set-Enable Register associated with the RTC\_ALM interrupt, an alarm interrupt is generated in the event of the alarm, enabling the alarm to be detected.

Writing 0 sets the IELSRn.IR flag associated with the RTC\_ALM interrupt to 0. If interrupt is enabled, the Interrupt Set-Pending/Clear-Pending Register associated with the RTC\_ALM interrupt is cleared automatically after exiting the interrupt handler. Otherwise, write 1 to the Interrupt Clear-Pending Register associated with the RTC\_ALM interrupt to clear it.

When the counter and the alarm time match in a low power state, the MCU returns from the low power state.

Note 1. For any bits in the ENB bits that are set to 1, the values in the corresponding positions in the alarm registers from the following registers are compared with the corresponding bits of the counted values.

Counter registers: RSECCNT, RMINCNT, RHRCNT, RWKCNT, RDAYCNT, RMONCNT, RYRCNT

Alarm registers: RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, RYRAREN

### 24.3.7 Procedure for Disabling Alarm Interrupt

Figure 24.8 shows the procedure for disabling the enabled alarm interrupt request.

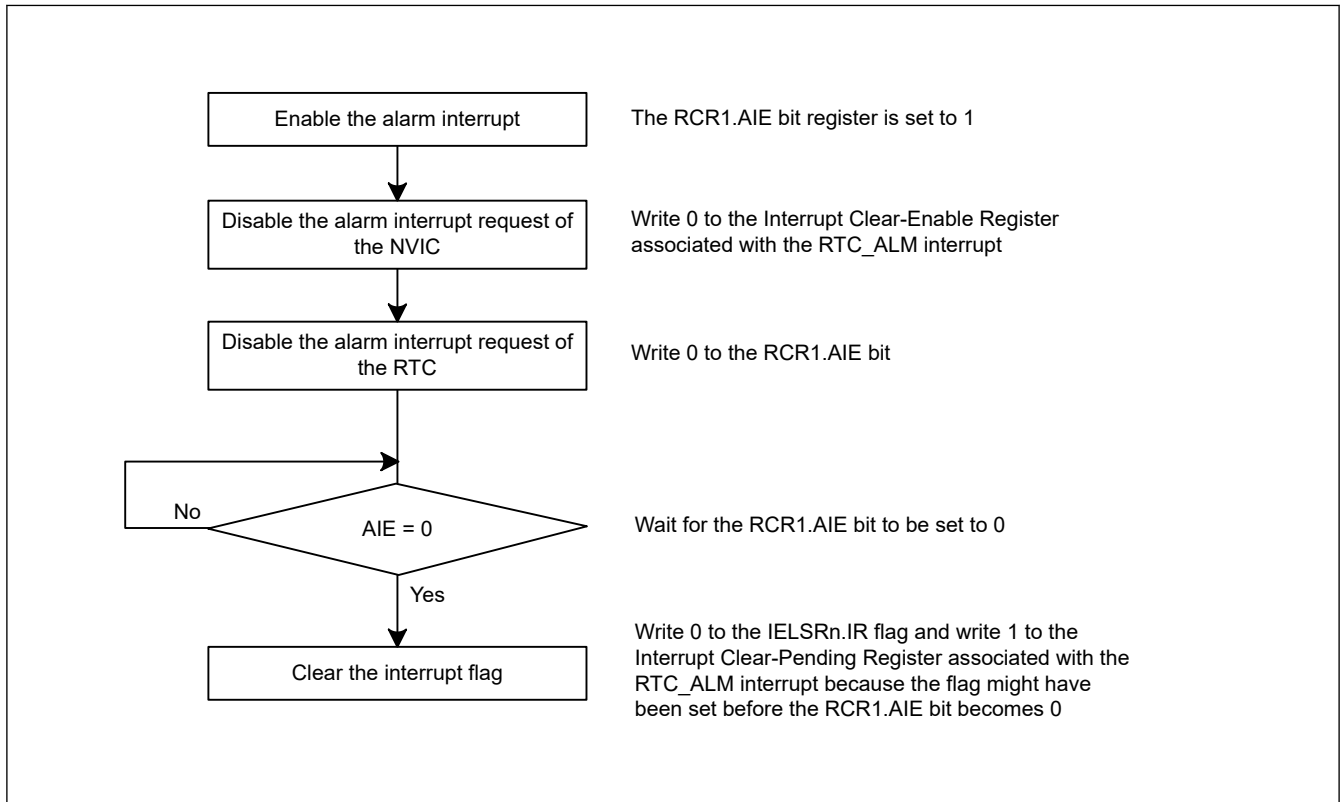


Figure 24.8 Procedure for disabling alarm interrupt request

### 24.3.8 Time Error Adjustment Function

The time error adjustment function is used to correct errors, running fast or slow, in the time caused by variation in the precision of oscillation by the sub-clock oscillator. Because 32768 cycles of the sub-clock oscillator constitute 1 second of operation when the sub-clock oscillator is selected, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low.

The time error adjustment functions include:

- Automatic adjustment
- Adjustment by software

Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

#### 24.3.8.1 Automatic adjustment

Enable automatic adjustment by setting the RCR2.AADJE bit to 1.

Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the RCR2.AADJP bit elapses.

**(1) Example 1: Sub-clock oscillator running at 32.769 kHz****Adjustment procedure**

When the sub-clock oscillator is running at 32.769 kHz, 1 second elapses every 32769 clock cycles. The RTC is meant to run at 32768 clock cycles, so the clock runs fast by 1 clock cycle every second. The time on the clock is fast by 60 clock cycles per minute, so adjustment can take the form of setting the clock back by 60 cycles every minute.

**Register settings when RCR2.CNTMD = 0:**

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler)
- RADJ.ADJ[5:0] = 60 (0x3C)

**(2) Example 2: Sub-clock oscillator running at 32.766 kHz****Adjustment procedure**

When the sub-clock oscillator is running at 32.766 kHz, 1 second elapses every 32766 clock cycles. The RTC is meant to run at 32768 clock cycles, so the clock runs slow by 2 clock cycles every second. The time on the clock is slow by 20 clock cycles every 10 seconds, so adjustment can take the form of setting the clock forward by 20 cycles every 10 seconds.

**Register settings when RCR2.CNTMD = 0:**

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler)
- RADJ.ADJ[5:0] = 20 (0x14)

**(3) Example 3: Sub-clock oscillator running at 32.764 kHz****Adjustment procedure**

When the sub-clock oscillator is running at 32.764 kHz, 1 second elapses on 32764 clock cycles. Because the RTC operates for 32768 clock cycles as 1 second, the clock is delayed for 4 clock cycles per second. In 8 seconds, the delay is 32 clock cycles, therefore correction can be made by advancing the clock 32 clock cycles every 8 seconds.

**Register settings when RCR2.CNTMD = 1:**

- RCR2.AADJP = 1 (adjustment every 8 seconds)
- RADJ.PMADJ[1:0] = 01b (adjustment is performed by the addition to the prescaler)
- RADJ.ADJ[5:0] = 32 (0x20)

**24.3.8.2 Adjustment by software**

Enable adjustment by software by setting the RCR2.AADJE bit to 0. Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register on execution of a write instruction to the RADJ register.

**(1) Example 1: Sub-clock oscillator running at 32.769 kHz****Adjustment procedure**

When the sub-clock oscillator is running at 32.769 kHz, 1 second elapses every 32769 clock cycles. The RTC is meant to run at 32768 clock cycles, so the clock runs fast by 1 clock cycle every second. The time on the clock is fast by 1 clock cycle per second, so adjustment can take the form of setting the clock back by 1 cycle every second.

**Register settings**

- RADJ.PMADJ[1:0] = 10b (adjustment is performed by the subtraction from the prescaler)
- RADJ.ADJ[5:0] = 1 (0x01)  
This is written to the RADJ register once per 1-second interrupt.

### 24.3.8.3 Procedure to change the mode of adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

To change adjustment by software to automatic adjustment:

1. Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
2. Set the RCR2.AADJE bit to 1 (automatic adjustment is enabled).
3. Use the RCR2.AADJP bit to select the period of adjustment.
4. In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time error adjustment.

To change automatic adjustment to adjustment by software:

1. Set the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).
2. Set the RCR2.AADJE bit to 0 (adjustment by software is enabled).
3. Proceed with the adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time error adjustment at the wanted time. After that, the time is adjusted every time a value is written to the RADJ register.

### 24.3.8.4 Procedure to stop adjustment

Stop the adjustment by setting the RADJ.PMADJ[1:0] bits to 00b (adjustment is not performed).

### 24.3.9 Capturing the time

The RTC is capable of storing the month, date, hour, minute and second/binary counters 3 to 0 by detecting an edge of a signal on a time capture event input pin in calendar count mode or binary count mode.

A noise filter can also be used on a time capture event input pin. If the noise filter is enabled, the RTCCRn.TCST bit is set to 1 when the input level on the pin matches three times.

The noise filter can be switched on or off for each of the time capture event input pins. Set VBTICTLR.VCHnIEN (n = 0 to 2) to 1 to enable the RTCICn input. Operation when the noise filter is off is shown in Figure 24.9 and operation when the noise filter is on is shown in Figure 24.10.

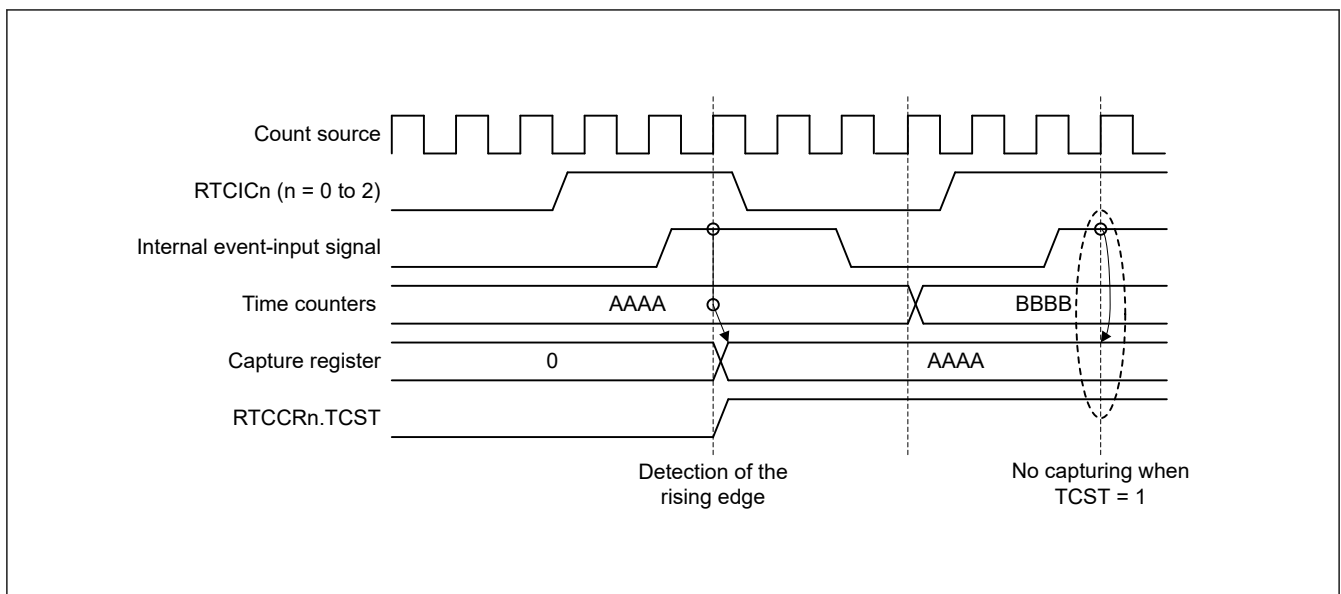


Figure 24.9 Timing of a time capture operation with the noise filter off

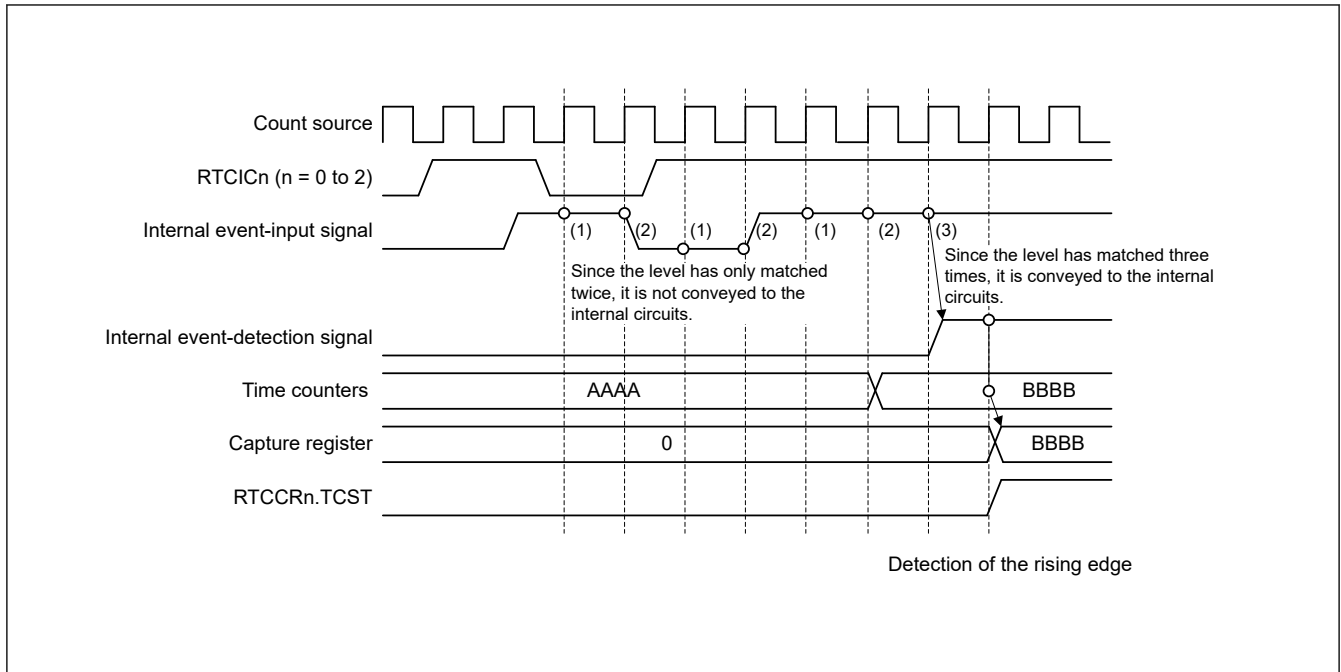


Figure 24.10 Timing of a time capture operation with the noise filter on

## 24.4 Interrupt Sources

The RTC has three interrupt sources, as listed in [Table 24.3](#).

Table 24.3 RTC interrupt sources

Name	Interrupt source
RTC_ALM	Alarm interrupt
RTC_PRD	Periodic interrupt
RTC_CUP	Carry interrupt

### (1) Alarm interrupt (RTC\_ALM)

This interrupt is generated based on the comparison result between the alarm registers and RTC counters. For details, see [section 24.3.6. Alarm Function](#).

Because there is a possibility that the interrupt flag might be set to 1 when the settings of the alarm registers match the clock counters, wait for the alarm time settings to be confirmed and clear the IELSRn.IR flag and the interrupt Set-Pending Register associated with the RTC\_ALM interrupt to 0 again after modifying values of the alarm registers. After the interrupt flag for the alarm interrupt is set to 1 and the state is returned to mismatching of the alarm registers and clock counters, the flag is not 1 again until there is another match or the values of the alarm registers are modified again.



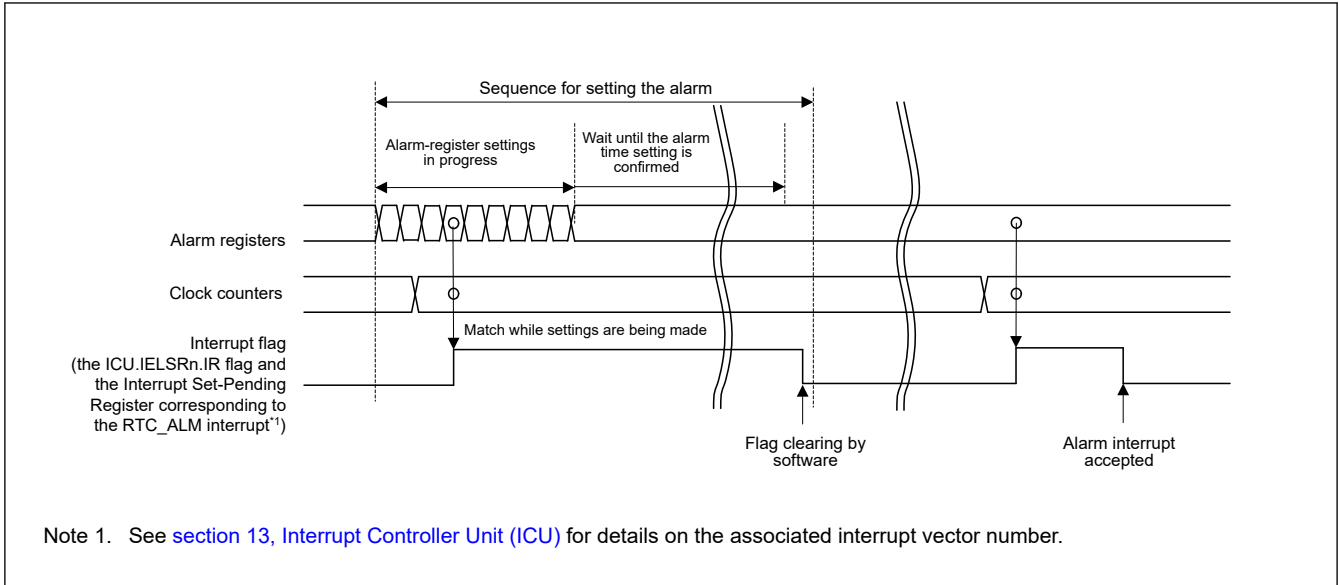


Figure 24.11 Timing for the alarm interrupt (RTC\_ALM)

(2) Periodic interrupt (RTC\_PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected in the RCR1.PES[3:0] bits.

(3) Carry interrupt (RTC\_CUP)

This interrupt is generated when a carry to the second counter/binary counter 0 occurred or a carry to the R64CNT counter occurred during read access to the 64-Hz counter.

Figure 24.12 shows the timing of the carry interrupt (RTC\_CUP).

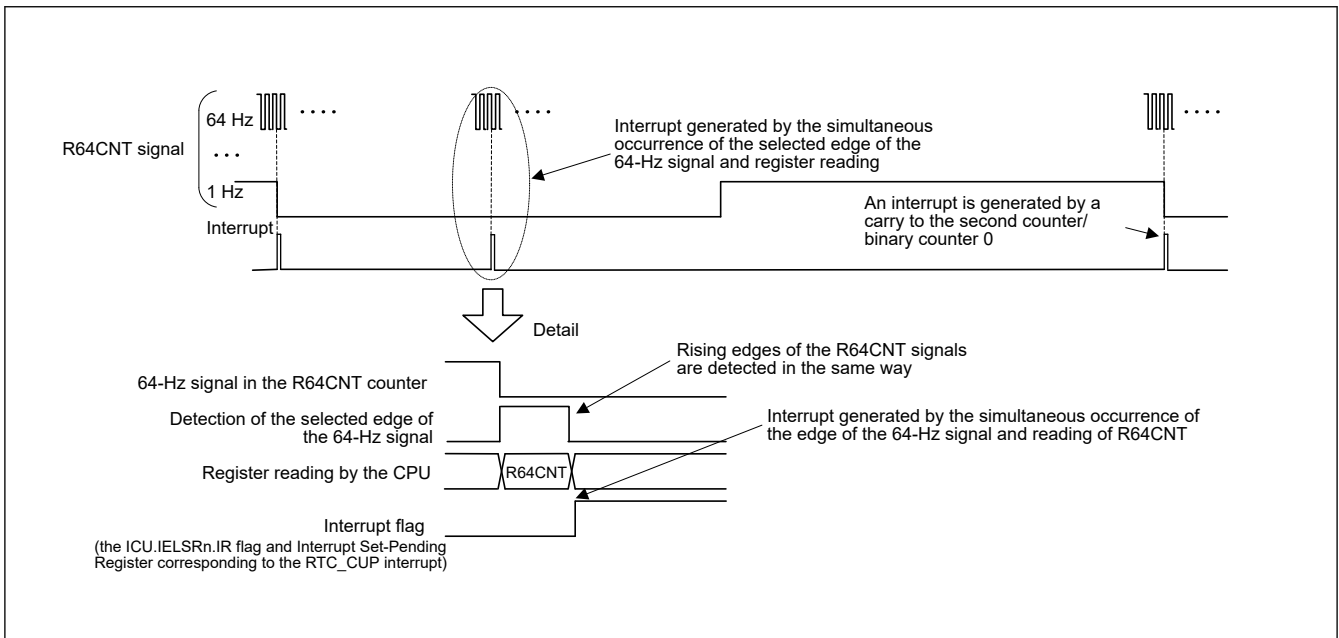


Figure 24.12 Timing for the carry interrupt (RTC\_CUP)

24.5 Event Link Output

The RTC generates periodic event output (RTC\_PRD) event signal for the ELC that can be used to initiate operations by other modules selected in advance.

The periodic event signal is output at the interval selected from 1/256, 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2, 1, and 2 seconds by setting the RCR1.PES[3:0] bits.

The event generation period immediately after the event generation is selected is not guaranteed.

**Note:** If event linking from the RTC is used, only set the ELC after setting the RTC, for example initialization and time settings. Setting the RTC after the ELC can lead to output of unexpected event signals.

### 24.5.1 Interrupt Handling and Event Linking

The RTC has a bit to enable or disable periodic interrupts. An interrupt request signal is output to the CPU when an interrupt source is generated while the associated enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal through the ELC when an interrupt source is generated, regardless of the setting of the associated interrupt enable bit.

**Note:** Although alarm and periodic interrupts can still be output during Software Standby or Deep Software Standby mode, the periodic event signals for the ELC are not output.

## 24.6 Usage Notes

### 24.6.1 Register Writing during Counting

The following registers should not be written to during counting, that is, while the RCR2.START bit is 1:

- RSECCNT/BCNT0
- RMINCNT/BCNT1
- RHRCNT/BCNT2
- RDAYCNT
- RWKCNT/BCNT3
- RMONCNT
- RYRCNT
- RCR1.RTCOS
- RCR2.RTCOE
- RCR2.HR24
- RFRL

The counter should be stopped before writing to any of these registers.

### 24.6.2 Use of Periodic Interrupts

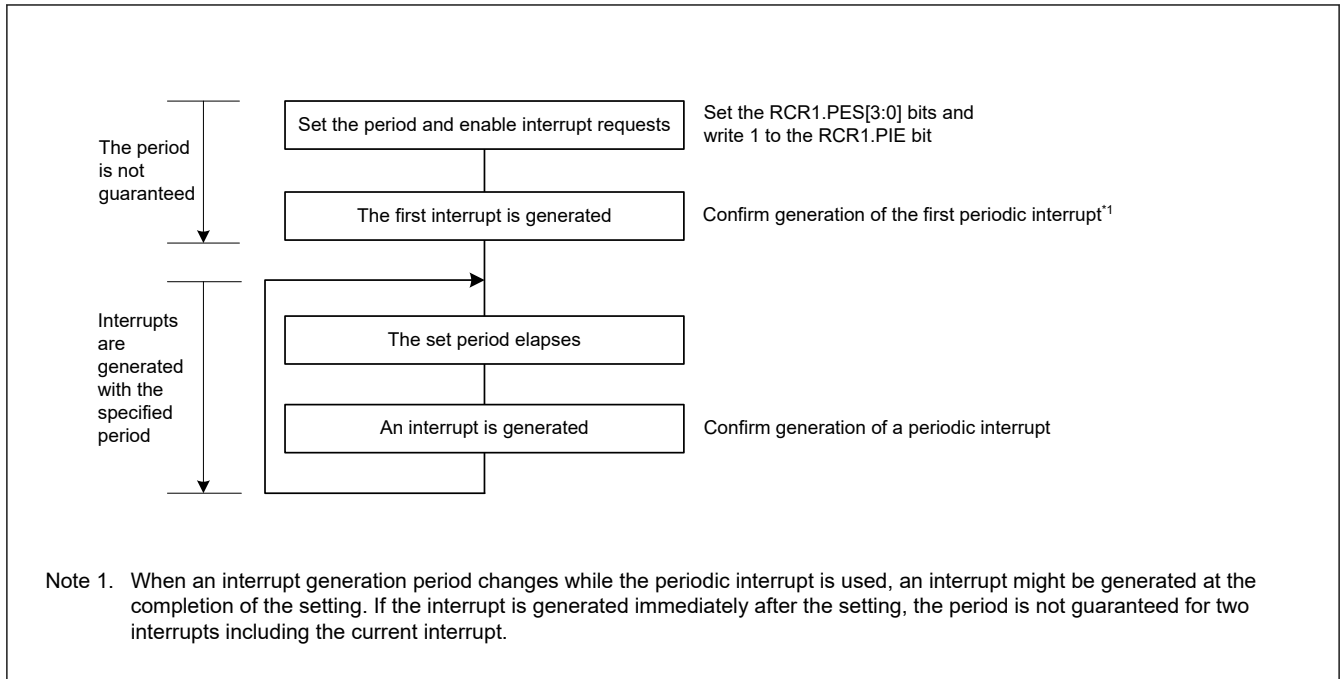
Figure 24.13 shows the procedure for using periodic interrupts.

The generation and period of the periodic interrupt can be changed by setting the RCR1.PES[3:0] bits. However, because the prescaler R64CNT and RSECCNT/BCNT0 are used to generate interrupts, the interrupt period is not guaranteed immediately after setting the RCR1.PES[3:0] bits.

In addition, any of the following operation can affect the interrupt period:

- Stopping/restarting or resetting counter operation
- Reset by RTC software
- 30-second adjustment by changing the RCR2 value

When the time error adjustment function is used, the interrupt generation period after adjustment is added or subtracted based on the adjustment value.



**Figure 24.13 Using the periodic interrupt function**

### 24.6.3 RTCOUT (1-Hz/64-Hz) Clock Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of RTCOUT (1-Hz/64-Hz) output. When the time error adjustment function is used, the period of RTCOUT (1-Hz/64-Hz) output after adjustment is added or subtracted based on the adjustment value.

### 24.6.4 Transitions to Low Power Modes after Setting Registers

A transition to a low power state (Software Standby mode, Deep Software Standby mode, or battery backup state) during a write to an RTC register might corrupt the value of the register. After setting the register, confirm that the setting is in place before initiating a transition to a low power state.

### 24.6.5 Notes on Writing to and Reading from Registers

- When reading a counter register such as the second counter after writing to the counter register, follow the procedure in [section 24.3.5. Reading 64-Hz Counter and Time](#).
- The value written to the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, RCR4 register, or frequency register is reflected when fourth read operations are performed after writing.
- The values written to the RCR1.CIE, RCR1.RTCOS, and RCR2.RTCOE bits can be read immediately after writing.
- To read the value from the timer counter after returning from a reset or a period in Software Standby mode, Deep Software Standby mode, or battery backup state, wait for 1/128 second while the clock is operating (RCR2.START bit = 1).
- After a reset is generated, write to the RTC register after 6 cycles of the count source clock have elapsed.

### 24.6.6 Changing the Count Mode

When changing the count mode (calendar count mode/binary count mode), set the RCR2.START bit to 0, stop the counting operation, then start it again from the initial setting. For details on the initial setting, see [section 24.3.1. Outline of Initial Settings of Registers after Power On](#).

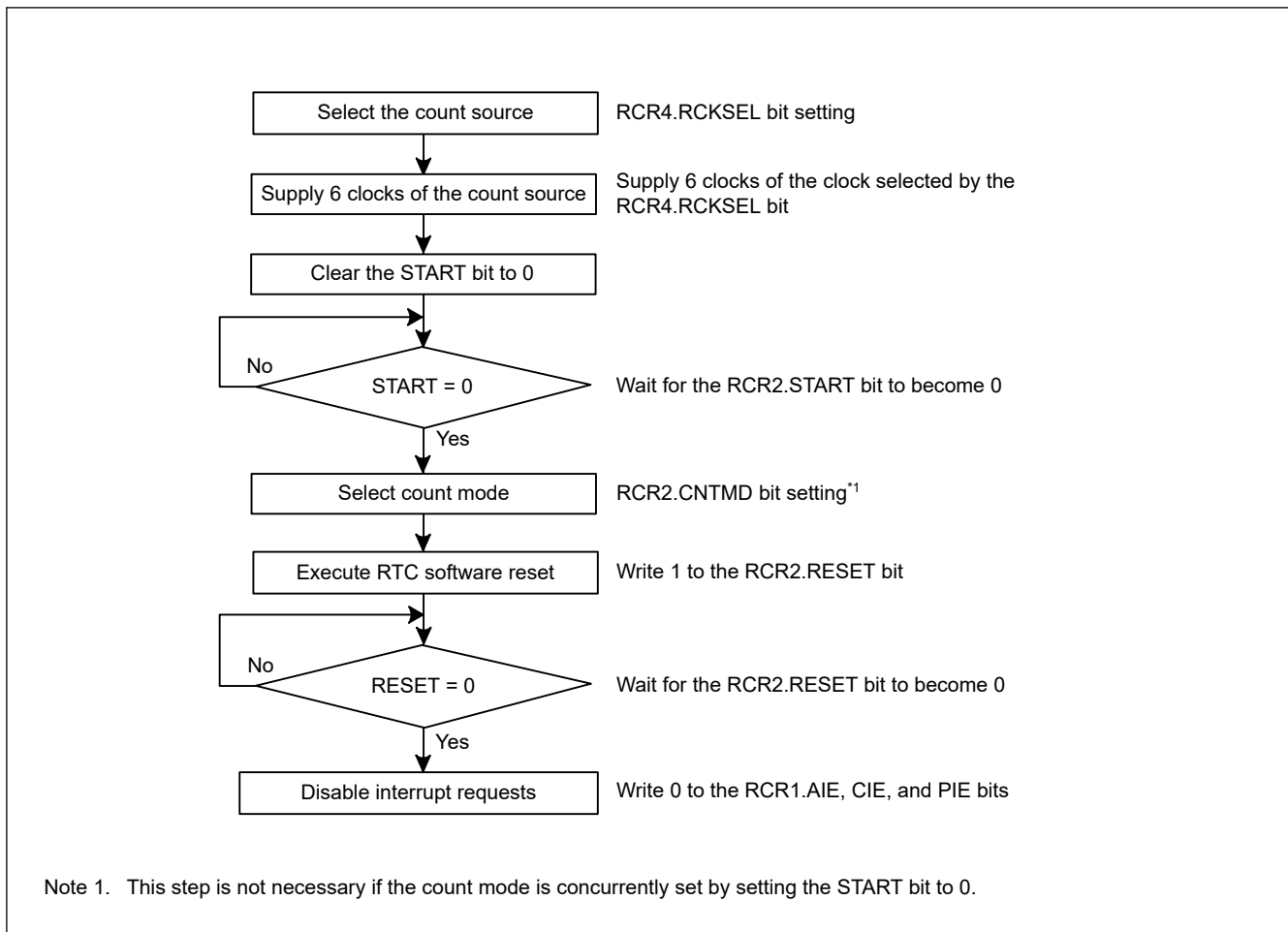
### 24.6.7 Initialization Procedure When the RTC Is Not to Be Used

Registers in the RTC are not initialized by a reset. Depending on the initial state, the generation of an unintentional interrupt request or operation of the counter might lead to increased power consumption.

For applications that do not require a realtime clock, initialize the registers by following the initialization procedure shown in [Figure 24.14](#).

Alternatively, when the sub-clock oscillator is not used as the system clock source, realtime clock, operating clock of each peripheral modules or FLL function of HOCO, the counter can be stopped by writing 0 (sub-clock oscillator is selected) to the RCR4.RCKSEL bit and stopping the sub-clock oscillator. To stop the sub-clock oscillator, write 1 to the SOSCCR.SOSTP bit.

For details on the setting of the SOSCCR.SOSTP bit, see [section 8, Clock Generation Circuit](#).



**Figure 24.14 Initialization procedure**

### 24.6.8 When Switching Source Clock

When switching a clock source by changing SCKSCR.CKSEL[2:0], the clock output from the selector stops for 4 cycles of the switched clock. If the RTC periodical interrupt or RTC periodical event output was generated at this time, the interrupt or event is invalid.

### 24.6.9 When writing to the RTC registers

If VCC voltage of the MCU is lower than 1.8V and write access to the RTC registers is performed, read access should be performed at least once after one write access is performed, or for the consecutive write access, leave an interval of 167ns for each access.

## 25. Watchdog Timer (WDT)

### 25.1 Overview

The Watchdog Timer (WDT) is a 14-bit down counter that can be used to reset the MCU when the counter underflows because the system has run out of control and is unable to refresh the WDT. In addition, the WDT can be used to generate a non-maskable interrupt or an underflow interrupt.

Table 25.1 lists the WDT specifications and Figure 25.1 shows a block diagram.

**Table 25.1 WDT specifications**

Parameter	Specifications
Count source*1	Peripheral clock (PCLKB)
Clock division ratio	Division by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Condition for starting the counter	<ul style="list-style-type: none"> <li>Auto start mode: Counting automatically starts after a reset or after an underflow or refresh error occurs</li> <li>Register start mode: Counting is started with a refresh by writing to the WDTRR register</li> <li>Only secure developer can select Auto-start mode or Register-start mode</li> </ul>
Conditions for stopping the counter	<ul style="list-style-type: none"> <li>Reset (the down-counter and other registers return to their initial values)</li> <li>A counter underflows or a refresh error is generated</li> </ul>
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Watchdog timer reset sources	<ul style="list-style-type: none"> <li>Down-counter underflows</li> <li>Refreshing outside the refresh-permitted period (refresh error)</li> </ul>
Non-maskable interrupt/interrupt sources	<ul style="list-style-type: none"> <li>Down-counter underflows</li> <li>Refreshing outside the refresh-permitted period (refresh error)</li> </ul>
Reading of the counter value	The down-counter value can be read by the WDTSR register
Event link function (output)	<ul style="list-style-type: none"> <li>Down-counter underflow event output</li> <li>Refresh error event output</li> </ul>
Output signal (internal signal)	<ul style="list-style-type: none"> <li>Reset output</li> <li>Interrupt request output</li> <li>CPU Sleep mode or CPU Deep Sleep mode count stop control output</li> </ul>
TrustZone Filter	Security and Privilege attribution can be set

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB)  $\geq 4 \times$  (the frequency of the count clock source after division).

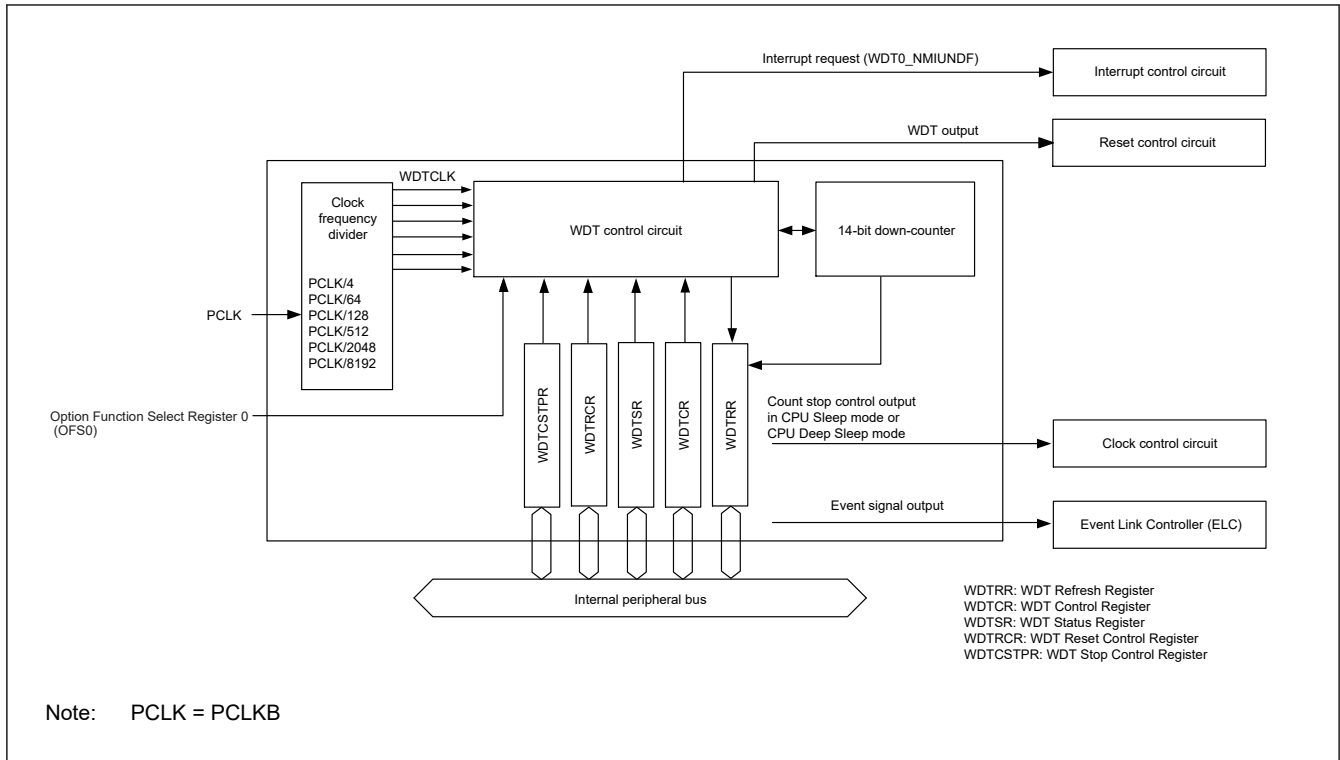


Figure 25.1 WDT block diagram

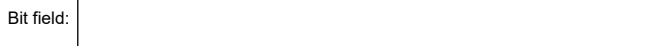
## 25.2 Register Descriptions

### 25.2.1 WDTRR : WDT Refresh Register

Base address: WDT0 = 0x4020\_2600  
 WDT0\_NS = 0x5020\_2600

Offset address: 0x00

Bit position: 7 0



Value after reset: 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
7:0	n/a	The down-counter is refreshed by writing 0x00 and then writing 0xFF to this register.	R/W

Note: S-TYPE-3, P-TYPE-3

The WDTRR register refreshes the down-counter of the WDT.

The down-counter of the WDT is refreshed by writing 0x00 and then writing 0xFF to WDTRR register (refresh operation) within the refresh-permitted period.

After the down-counter is refreshed, it starts counting down from the value selected by setting the WDT Timeout Period Select bits (OFS0.WDTPOPS[1:0]) in the Option Function Select Register 0 in auto start mode. In register start mode, counting down starts from the value selected by setting the Timeout Period Select bits (WDTCR.TOPS[1:0]) in the WDT Control Register.

When 0x00 is written, the read value is 0x00. When a value other than 0x00 is written, the read value is 0xFF. For details of the refresh operation, see [section 25.3.3. Refresh Operation](#).

## 25.2.2 WDTCR : WDT Control Register

Base address: WDT0 = 0x4020\_2600  
WDT0\_NS = 0x5020\_2600

Offset address: 0x02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RPSS[1:0]	—	—	RPES[1:0]	CKS[3:0]			—	—	TOPS[1:0]				
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Function	R/W
1:0	TOPS[1:0]	Timeout Period Select 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	CKS[3:0]	Clock Division Ratio Select 0 0 0 1: PCLKB/4 0 1 0 0: PCLKB/64 1 1 1 1: PCLKB/128 0 1 1 0: PCLKB/512 0 1 1 1: PCLKB/2048 1 0 0 0: PCLKB/8192 Others: Setting prohibited	R/W
9:8	RPES[1:0]	Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (do not specify window end position).	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	RPSS[1:0]	Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (do not specify window start position).	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The WDTCR register is used to set the clock division ratio, and window start and end positions for refresh, and the timeout period until the down-counter underflows in register start mode.

Some constraints apply to writes to the WDTCR register. For details, see [section 25.3.2. Controlling Writes to the WDTCR, WDTRCR, and WDTCTPR Registers](#).

In auto start mode, the settings in the WDTCR register are disabled, and the settings in the Option Function Select Register 0 (OFS0) are enabled. The settings for the WDTCR register can also be made in the OFS0 register. For details, see [section 25.3.8. Association between Option Function Select Register 0 \(OFS0\) and WDT Registers](#).

### TOPS[1:0] bits (Timeout Period Select)

The TOPS[1:0] bits select the timeout period, the period until the down-counter underflows, from 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified in the CKS[3:0] bits as 1 cycle. After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the number of PCLKB cycles until the counter underflows.

[Table 25.2](#) lists the relationship between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKB cycles.

**Table 25.2** Timeout period settings

CKS[3:0] bits	TOPS[1:0] bits	Clock division ratio	Timeout period (number of cycles)	PCLKB clock cycles
0x1	00b	PCLKB/4	1024	4096
	01b		4096	16384
	10b		8192	32768
	11b		16384	65536
0x4	00b	PCLKB/64	1024	65536
	01b		4096	262144
	10b		8192	524288
	11b		16384	1048576
0xF	00b	PCLKB/128	1024	131072
	01b		4096	524288
	10b		8192	1048576
	11b		16384	2097152
0x6	00b	PCLKB/512	1024	524288
	01b		4096	2097152
	10b		8192	4194304
	11b		16384	8388608
0x7	00b	PCLKB/2048	1024	2097152
	01b		4096	8388608
	10b		8192	16777216
	11b		16384	33554432
0x8	00b	PCLKB/8192	1024	8388608
	01b		4096	33554432
	10b		8192	67108864
	11b		16384	134217728

**CKS[3:0] bits (Clock Division Ratio Select)**

The CKS[3:0] bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from the PCLKB divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, this allows the WDT to be configured to a count period between 4096 and 134217728 PCLKB clock cycles.

**RPES[1:0] bits (Window End Position Select)**

The RPES[1:0] bits specify the window end position that indicates the refresh-permitted period. 75%, 50%, 25%, or 0% of the timeout period can be selected for the window end position. Set the window end position to a value less than the value for the window start position (window start position > window end position). If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

**RPSS[1:0] bits (Window Start Position Select)**

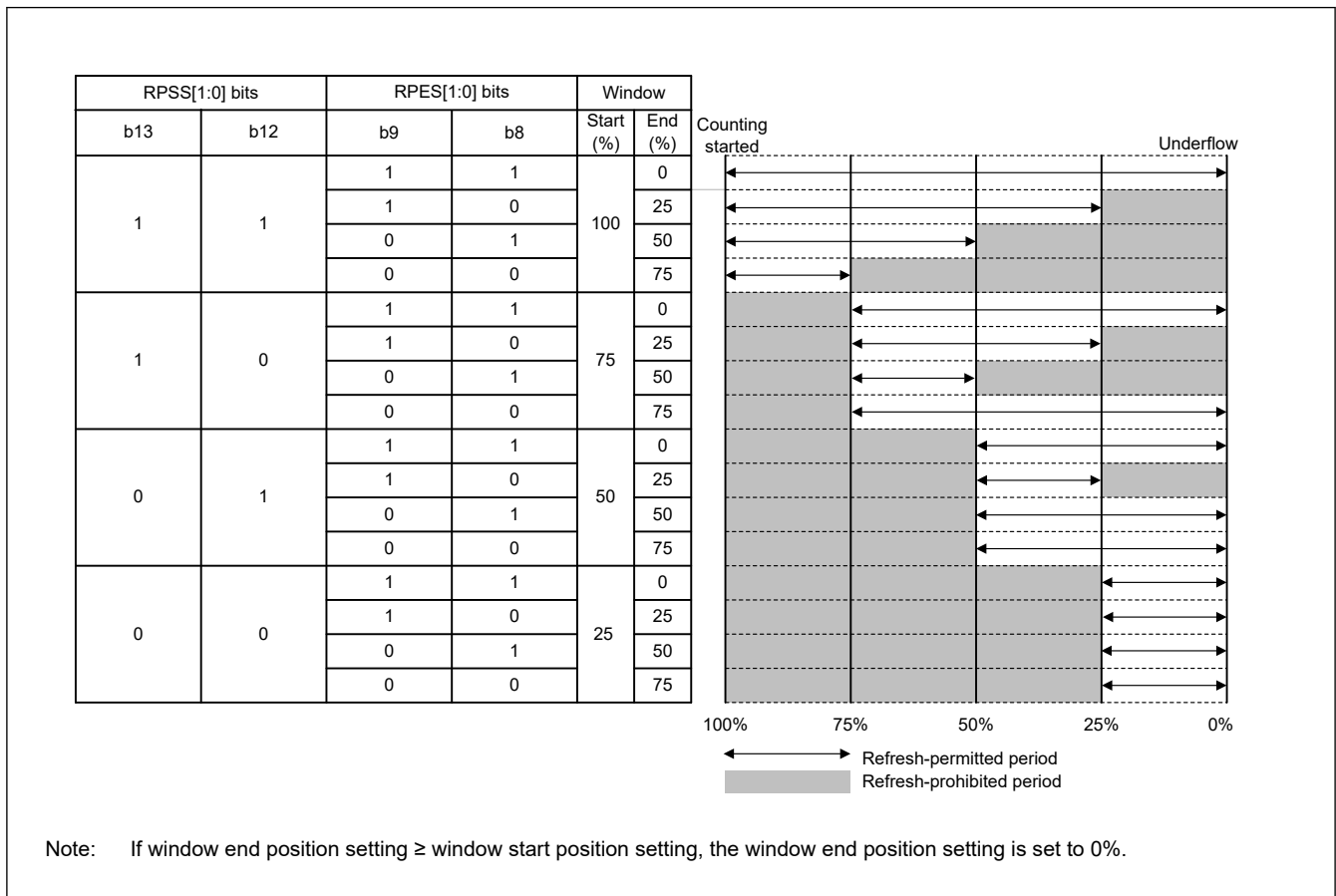
The RPSS[1:0] bits specify the window start position that indicates the refresh-permitted period. 100%, 75%, 50%, or 25% of the timeout period can be selected for the window start position. Set the window start position to a value greater than the value for the window end position. If the window start position is set to a value less than or equal to the window end position, the window start position setting is enabled and the window end position is set to 0%.

[Table 25.3](#) lists the counter values for the window start and end positions, and [Figure 25.2](#) shows the refresh-permitted period set in the RPSS[1:0], RPES[1:0], and TOPS[1:0] bits.



**Table 25.3 Relationship between the timeout period and window start and end counter values**

TOPS[1:0]	Timeout period		Window start and end counter value			
	Cycles	Counter value	100%	75%	50%	25%
00b	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
01b	4096	0x0FFF	0x0FFF	0x0BFF	0x07FF	0x03FF
10b	8192	0x1FFF	0x1FFF	0x17FF	0x0FFF	0x07FF
11b	16384	0x3FFF	0x3FFF	0x2FFF	0x1FFF	0x0FFF



**Figure 25.2 RPSS[1:0] and RPES[1:0] bits setting and refresh-permitted period**

### 25.2.3 WDTSR : WDT Status Register

Base address: WDT0 = 0x4020\_2600  
 WDT0\_NS = 0x5020\_2600

Offset address: 0x04

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	REFE F	UNDF F	CNTVAL[13:0]												
------------	-----------	-----------	--------------	--	--	--	--	--	--	--	--	--	--	--	--

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-Counter Value Value counted by the down-counter	R
14	UNDF	Underflow Flag 0: No underflow occurred 1: Underflow occurred	R/W <sup>1</sup>

Bit	Symbol	Function	R/W
15	REFEF	Refresh Error Flag 0: No refresh error occurred 1: Refresh error occurred	R/W <sup>1</sup>

Note: S-TYPE-3, P-TYPE-3

Note 1. Only 0 can be written to clear the flag.

The WDTSR register indicates the counter value of the down-counter and the status of whether an underflow or refresh error occurred in the down-counter.

### CNTVAL[13:0] bits (Down-Counter Value)

Read the CNTVAL[13:0] bits to confirm the value of the down-counter. The read value might differ from the actual count by 1.

### UNDFE flag (Underflow Flag)

Read the UNDFE flag to confirm whether an underflow occurred in the counter. A value of 1 indicates that the down counter underflowed. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFE flag takes (N + 1) PCLKB cycles. In addition, clearing of the flag is ignored for (N + 1) PCLKB cycles after an underflow. N is specified in the WDTCCR.CKS[3:0] bits as follows:

- When WDTCCR.CKS[3:0] = 0x1, N = 4
- When WDTCCR.CKS[3:0] = 0x4, N = 64
- When WDTCCR.CKS[3:0] = 0xF, N = 128
- When WDTCCR.CKS[3:0] = 0x6, N = 512
- When WDTCCR.CKS[3:0] = 0x7, N = 2048
- When WDTCCR.CKS[3:0] = 0x8, N = 8192

### REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error occurred, indicating that a refresh operation was performed during a prohibited period. A value of 1 indicates that a refresh error occurred. Write 0 to the flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFEF flag takes (N + 1) PCLKB cycles. In addition, clearing of the flag is ignored for (N + 1) PCLKB cycles after a refresh error. N is specified in the WDTCCR.CKS[3:0] bits as follows:

- When WDTCCR.CKS[3:0] = 0x1, N = 4
- When WDTCCR.CKS[3:0] = 0x4, N = 64
- When WDTCCR.CKS[3:0] = 0xF, N = 128
- When WDTCCR.CKS[3:0] = 0x6, N = 512
- When WDTCCR.CKS[3:0] = 0x7, N = 2048
- When WDTCCR.CKS[3:0] = 0x8, N = 8192

## 25.2.4 WDTRCR : WDT Reset Control Register

Base address: WDT0 = 0x4020\_2600  
WDT0\_NS = 0x5020\_2600

Offset address: 0x06

Bit position: 7 6 5 4 3 2 1 0

Bit field:	RSTIR QS	—	—	—	—	—	—
------------	-------------	---	---	---	---	---	---

Value after reset: 1 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	RSTIRQS	WDT Behavior Selection 0: Interrupt 1: Reset	R/W

Note: S-TYPE-3, P-TYPE-3

The WDTRCR register controls reset output by a WDT down-counter underflow or interrupt request output.

Some constraints apply to writes to the WDTRCR register. For details, see [section 25.3.2. Controlling Writes to the WDTCSR, WDTRCR, and WDTCSNPR Registers](#).

In auto start mode, the WDTRCR register settings are disabled, and the settings in the Option Function Select register 0 (OFS0) are enabled. The settings for the WDTRCR register can also be made for the OFS0 register. For details, see [section 25.3.8. Association between Option Function Select Register 0 \(OFS0\) and WDT Registers](#).

### 25.2.5 WDTCSNPR : WDT Count Stop Control Register

Base address: WDT0 = 0x4020\_2600  
WDT0\_NS = 0x5020\_2600

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SLCS TP	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	SLCSTP	CPU Sleep-Mode Count Stop Control Register 0: Disable count stop 1: Stop count on transition to CPU Sleep mode or CPU Deep Sleep mode	R/W

Note: S-TYPE-3, P-TYPE-3

The WDTCSNPR register controls whether to stop the WDT counter in CPU Sleep mode or CPU Deep Sleep mode. Some constraints apply to writes to the WDTCSNPR register. For details, see [section 25.3.2. Controlling Writes to the WDTCSR, WDTRCR, and WDTCSNPR Registers](#).

In auto start mode, the WDTCSNPR register settings are disabled, and the settings in the Option Function Select register 0 (OFS0) are enabled. The settings for the WDTCSNPR register can also be made for the OFS0 register. For details, see [section 25.3.8. Association between Option Function Select Register 0 \(OFS0\) and WDT Registers](#).

#### SLCSTP bit (CPU Sleep-Mode Count Stop Control Register)

The SLCSTP bit selects whether to stop counting on transition to CPU Sleep mode or CPU Deep Sleep mode.

### 25.2.6 Option Function Select Register 0 (OFS0)

For information on the OFS0 register, see [section 25.3.8. Association between Option Function Select Register 0 \(OFS0\) and WDT Registers](#).

## 25.3 Operation

### 25.3.1 Count Operation in each Start Mode

The WDT has two start modes:

- Auto start mode, in which counting automatically starts after a release from the reset state
- Register start mode, in which counting starts with a refresh by writing to the register.

In auto start mode, counting automatically starts after a release from the reset state according to the settings in the Option Function Select register 0 (OFS0) in the flash.

In register start mode, counting starts with a refresh by writing to the WDTRR register after the respective registers are set after a release from the reset state.

Select auto start mode or register start mode by setting the WDT Start Mode Select bit (OFS0.WDTSTRT) in the OFS0 register.

When the auto start mode is selected, the settings in the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSSTPR) are disabled while the settings in the OFS0 register are enabled.

When the register start mode is selected, the setting for the OFS0 register is disabled while the settings for the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSSTPR) are enabled.

### 25.3.1.1 Register start mode

When the WDT Start Mode Select bit (OFS0.WDTSTRT) is 1, register start mode is selected, the OFS0 register setting is invalid, and the WDT control register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSSTPR) are enabled.

After the reset state is released, set the following:

- Clock division ratio in the WDTCR register
- Window start and end positions in the WDTCR register
- Timeout period in the WDTCR register
- Reset output or interrupt request output in the WDTRCR register
- Counter stop control during transitions to CPU Sleep mode or CPU Deep Sleep mode in the WDTCSSTPR register

The WDT refresh register (WDTRR) refreshes the down counter. As a result, the downcount starts at the value set by the timeout period selection bit (WDTCR.TOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal or non-maskable interrupt request/interrupt request as long as counting continues. However, if the down-counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the WDT outputs the reset signal or a non-maskable interrupt request/interrupt request (WDT0\_NMIUNDF). Reset output or interrupt request output can be selected in the WDT Reset Interrupt Request Select bit (WDTRCR.RSTIRQS). The interrupt enabled for operating the NMI can be selected in the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 25.3 shows an example of operation under the following conditions:

- Register start mode (OFS0.WDTSTRT = 1)
- WDT reset interrupt request selection (WDTRCR.RSTIRQS = 1)
- The window start position is 75% (WDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (WDTCR.RPES[1:0] = 10b)

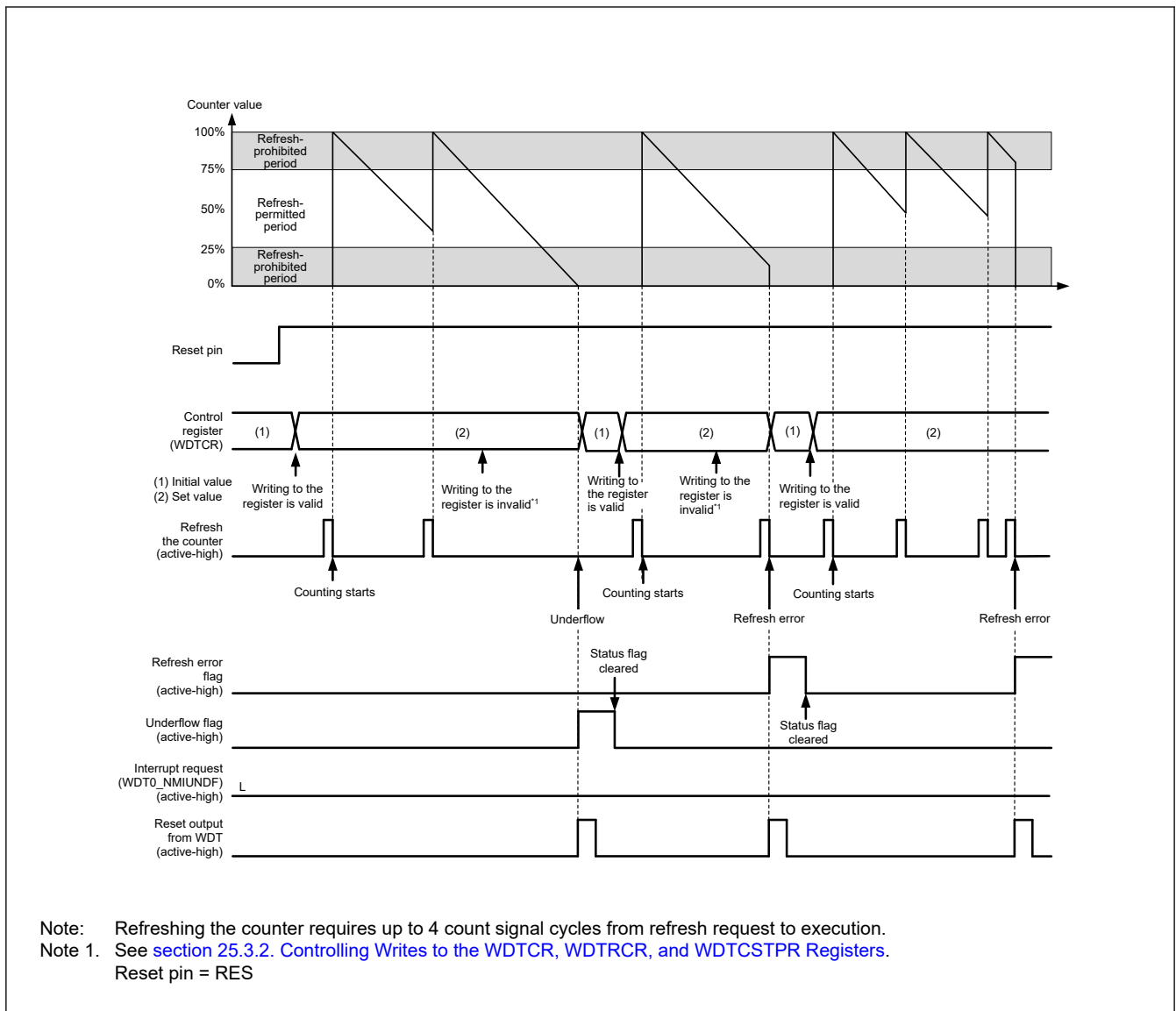


Figure 25.3 Operation example in register start mode

### 25.3.1.2 Auto start mode

When the WDT Start Mode Select bit (OFS0.WDTSTRT) in the Option Function Select Register 0 (OFS0) is 0, auto start mode is selected, the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), and WDT Count Stop Control Register (WDTCSSTPR) are disabled, and the settings in the OFS0 register are enabled.

Within the reset state, the setting values for the following in the Option Function Select Register 0 (OFS0) are set in the WDT registers:

- Clock division ratio
- Window start and end positions
- Timeout period
- Reset output or interrupt request
- Counter stop control during transition to CPU Sleep mode or CPU Deep Sleep mode

When the reset state is released, the down-counter automatically starts counting down from the value set in the WDT Timeout Period Select bits (OFS0.WDTPOPS[1:0]).

Thereafter, as long as the counter is refreshed in the refresh-permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The WDT does not output the reset signal or non-maskable interrupt

request/interrupt request (WDT0\_NMIUNDF) as long as the counting continues. However, if the down-counter underflows because refreshing of the down-counter is not possible due to a runaway program or if a refresh error occurs due to refreshing outside the refresh-permitted period, the WDT outputs the reset signal or non-maskable interrupt request/interrupt request (WDT0\_NMIUNDF).

After the reset signal or non-maskable interrupt request/interrupt request is generated, the counter reloads the timeout period after counting for 1 cycle. The value of the timeout period is set in the down-counter and counting restarts.

Reset output or interrupt request output can be selected by setting the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS). Non-maskable interrupt request or interrupt request can be selected in the WDT Underflow/Refresh Error Interrupt Enable bit (NMIER.WDTEN).

Figure 25.4 shows an example of operation (non-maskable interrupt) under the following conditions:

- Auto start mode (OFS0.WDTSTRT = 0)
- WDT behavior selection: interrupt (OFS0.WDTRSTIRQS = 0)
- Non-maskable Interrupt: WDT Underflow/Refresh Error Interrupt Enabled (NMIER.WDTEN = 1)
- The window start position is 75% (OFS0.WDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.WDTRPES[1:0] = 10b)

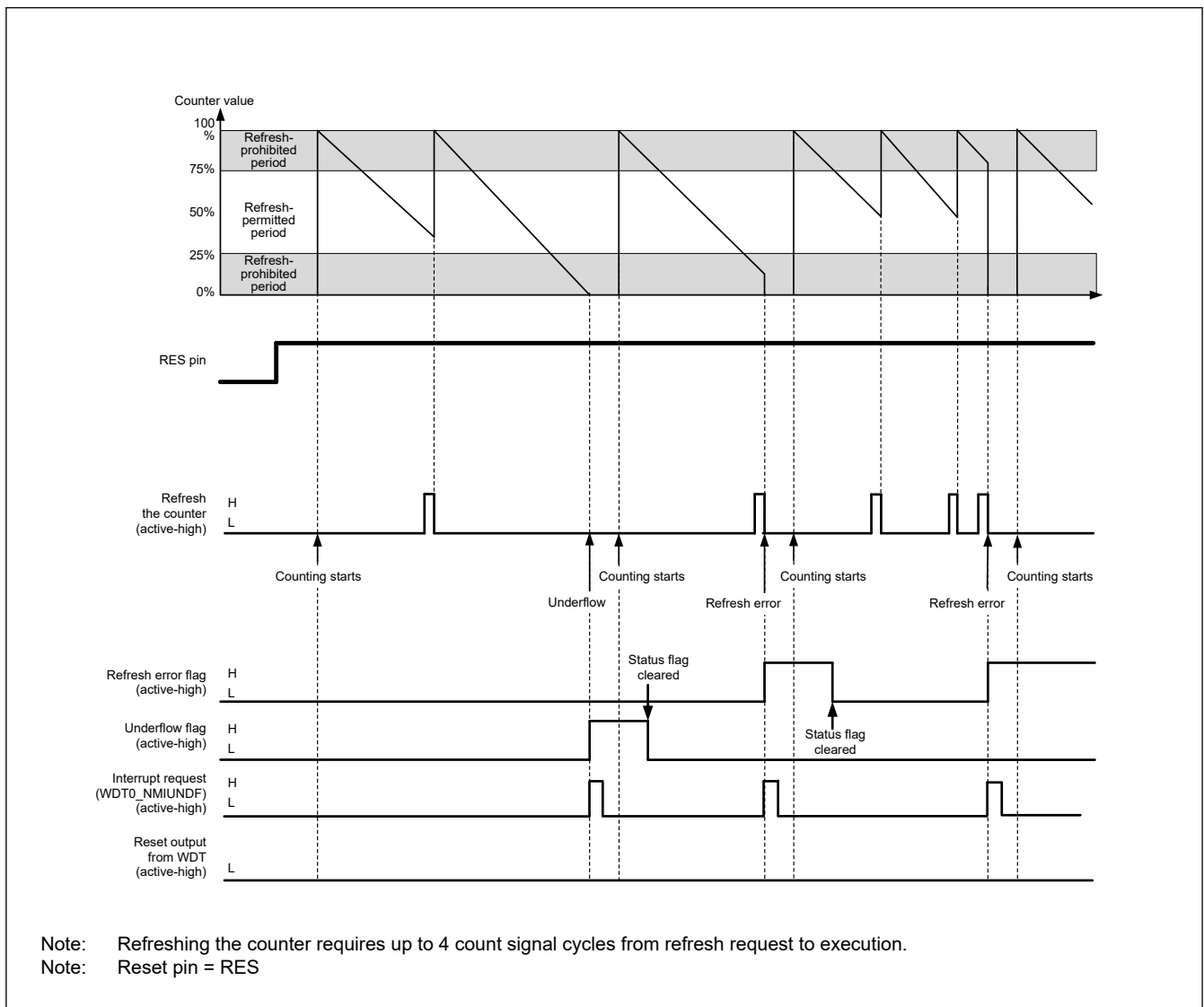


Figure 25.4 Operation example in auto start mode

### 25.3.2 Controlling Writes to the WDTCR, WDTRCR, and WDTCSSTPR Registers

Writing to the WDT Control Register (WDTCR), WDT Reset Control Register (WDTRCR), or WDT Count Stop Control Register (WDTCSSTPR) is possible once each between the release from the reset state and the first refresh operation.

After a refresh (counting starts) or a write to WDTCR, WDTRCR or WDTCSSTPR register, the protection signal in the WDT becomes 1 to protect WDTCR, WDTRCR and WDTCSSTPR register against subsequent write attempts. This protection is released by the reset source of the WDT. With other reset sources, the protection is not released.

Figure 25.5 shows control waveforms produced in response to writing to the WDTCR.

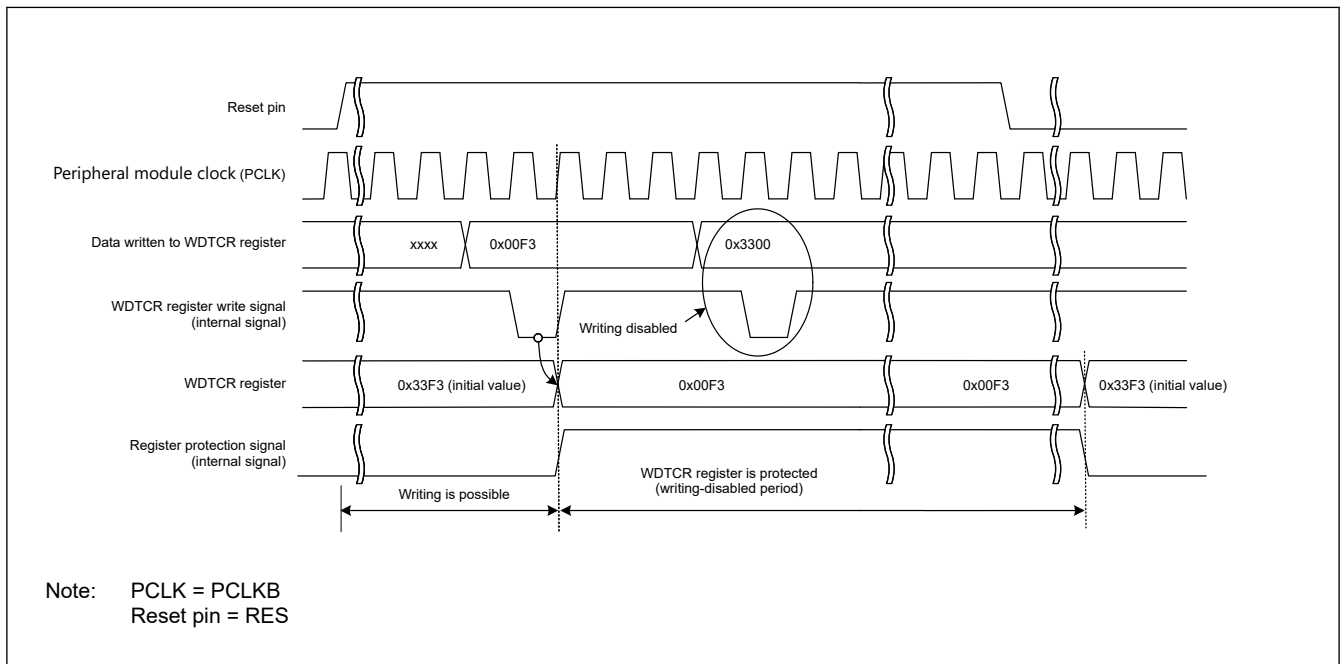


Figure 25.5 Control waveforms produced in response to writes to the WDTCR register

### 25.3.3 Refresh Operation

The down-counter is refreshed and starts counting operation on a write of the values 0x00 and then 0xFF to the WDT Refresh Register (WDTRR).

If a value other than 0xFF is written after 0x00, the down-counter is not refreshed.

If an invalid value is written, correct refreshing resumes on a write of 0x00 and then 0xFF to the WDTRR register.

Correct refreshing is also performed when a register other than WDTRR is accessed or WDTRR is read between writing 0x00 and writing 0xFF to WDTRR. Writes to refresh the counter must be made within the refresh-permitted period, and this is determined by the 0xFF write. For this reason, correct refreshing is performed even when 0x00 is written outside the refresh-permitted period.

[Example write sequences that are valid for refreshing the counter]

- 0x00 → 0xFF
- 0x00 ((n-1)th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from WDTRR → 0xFF

[Example write sequences that are invalid for refreshing the counter]

- 0x23 (a value other than 0x00) → 0xFF
- 0x00 → 0x54 (a value other than 0xFF)
- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF

After 0xFF is written to the WDT Refresh Register (WDTRR), refreshing the down-counter requires up to 4 cycles of the signal for counting. To meet this requirement, complete writing 0xFF to WDTRR 4 count cycles before the down-counter underflows.

Figure 25.6 shows the WDT refresh-operation waveforms when the clock division ratio is PCLKB/64.

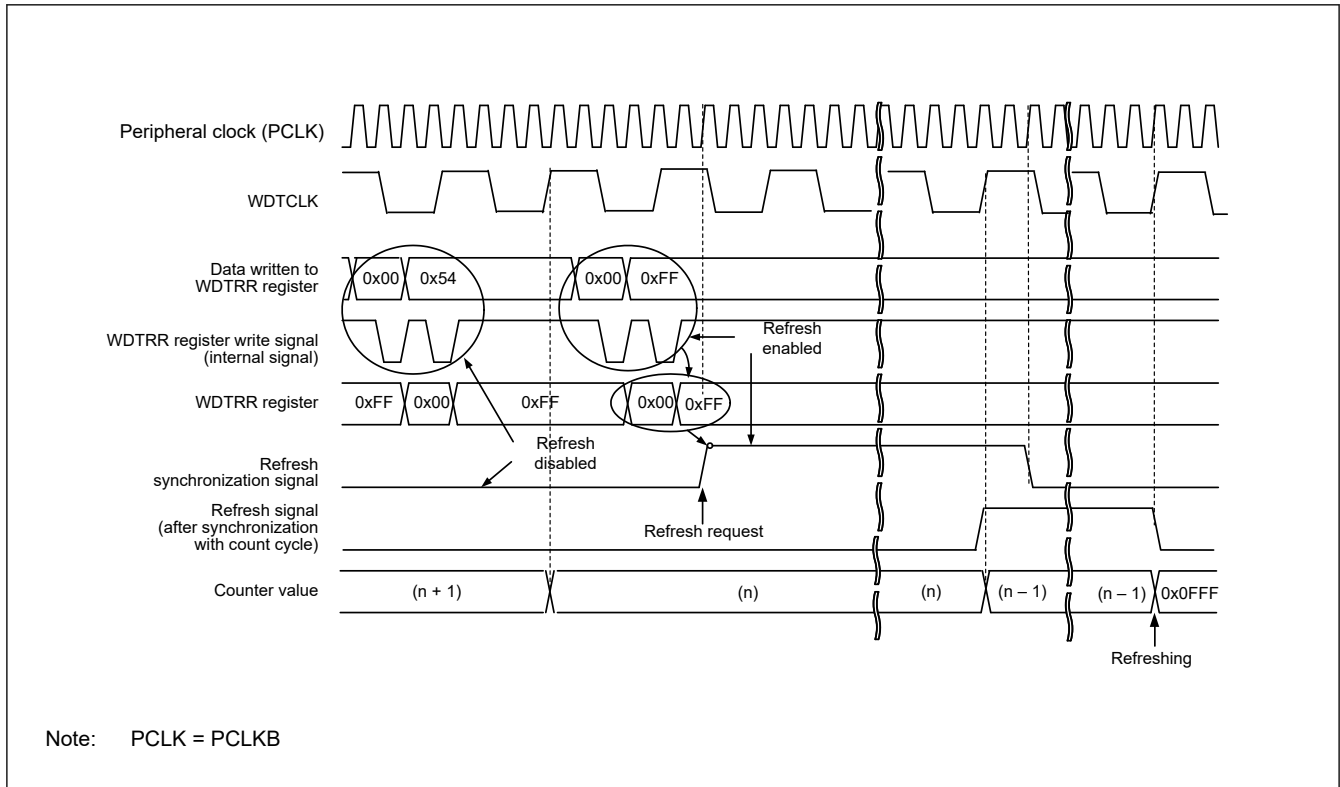


Figure 25.6 WDT refresh operation waveforms when WDTCR.CKS[3:0] = 0x4 and WDTCR.TOPS[1:0] = 01b

Note: When setting the refresh time, consider the oscillation accuracy of the clock sources of the PCLKB and WDTCLK. Set values which ensure that refreshing is possible even when the frequency varies in the range of error of the oscillation accuracy.

### 25.3.4 Status Flags

The refresh error (WDTSR.REFEF) and underflow (WDTSR.UNDF) flags retain the source of the interrupt request from the WDT. After a release from the interrupt request generation, read the WDTSR.REFEF and WDTSR.UNDF flags to check for the interrupt source. For each flag, writing 0 clears the bit. Writing 1 has no effect. Leaving the status flags unchanged does not affect operation. If the flags are not cleared at the next interrupt request from the WDT, the earlier interrupt source is cleared and the new interrupt source is written. For the time period between when 0 is written in each flag and when its value is reflected, see section 25.2.3. WDTSR : WDT Status Register.

### 25.3.5 Reset Output

When the Reset Interrupt Select bit (WDTSCR.RSTIRQS) is set to 1 in register start mode, or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 1 in auto start mode, a reset signal is output for 1 cycle count when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits set to 0) and stopped in that state after output of a reset signal. After the reset state is released and the program is restarted, the counter is set up again and counting down starts again with a refresh. In auto start mode, counting down starts automatically after the reset state is released.

### 25.3.6 Interrupt Sources

When the Reset Interrupt Select bit (WDTSCR.RSTIRQS) is set to 0 in register start mode or when the WDT Reset Interrupt Request Select bit (OFS0.WDTRSTIRQS) in the Option Function Select Register 0 (OFS0) is set to 0 in auto start



mode, an interrupt (WDT0\_NMIUNDF) signal is generated when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt or an interrupt. For details, see [section 13, Interrupt Controller Unit \(ICU\)](#).

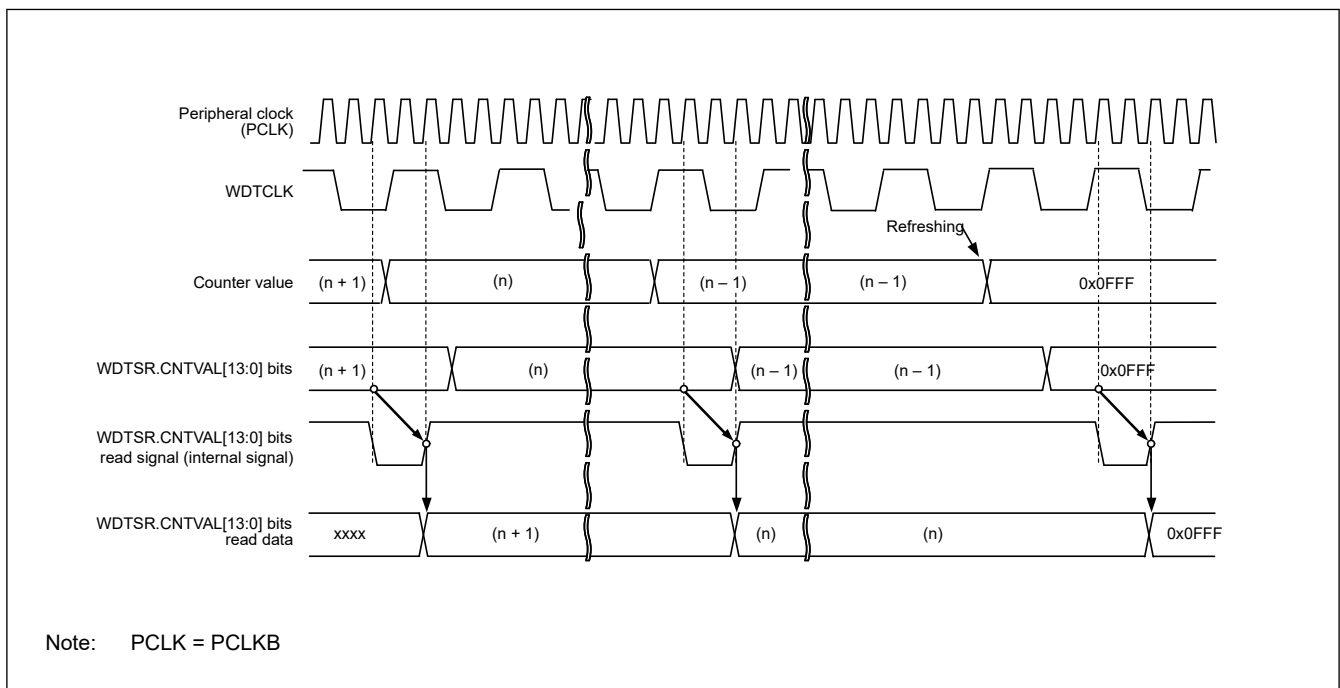
**Table 25.4 WDT interrupt source**

Name	Interrupt source	Interrupt to CPU	Start DMAC or DTC
WDT0_NMIUNDF	<ul style="list-style-type: none"> <li>Down-counter underflow</li> <li>Refresh error</li> </ul>	Possible	Not possible

### 25.3.7 Reading the Down-Counter Value

The WDT stores the counter value in the down-counter value bits (WDTSR.CNTVAL[13:0]) of the WDT Status Register. Check these bits to obtain the counter value. The read value of the down-counter might differ from the actual count by one.

[Figure 25.7](#) shows the processing for reading the WDT down-counter value when the clock division ratio is PCLKB/64.



**Figure 25.7 Processing for reading WDT down-counter value when WDTCR.CKS[3:0] = 0x4 and WDTCR.TOPS[1:0] = 01b**

### 25.3.8 Association between Option Function Select Register 0 (OFS0) and WDT Registers

[Table 25.5](#) lists the association between the Option Function Select Register 0 (OFS0) used in auto start mode, and the registers used in register start mode. For details on the Option Function Select Register 0 (OFS0), see [section 6.2.1. OFS0 : Option Function Select Register 0](#).

**Table 25.5 Association between Option Function Select Register 0 (OFS0) and the WDT registers (1 of 2)**

Control target	Function	OFS0 register (enabled in auto start mode) OFS0.WDTSTRT = 0	WDT registers (enabled in register start mode) OFS0.WDTSTRT = 1
Down-counter	Timeout period selection	OFS0.WDTPOPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]

**Table 25.5 Association between Option Function Select Register 0 (OFS0) and the WDT registers (2 of 2)**

Control target	Function	OFS0 register (enabled in auto start mode) OFS0.WDTSTRT = 0	WDT registers (enabled in register start mode) OFS0.WDTSTRT = 1
Reset output or interrupt request output	Select a reset interrupt request	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS
Count stop	CPU Sleep mode or CPU Deep Sleep mode count stop control	OFS0.WDTSTPCTL	WDTCSR.SLCSTP

## 25.4 Output to the Event Link Controller (ELC)

The WDT is capable of a link operation for the previously specified module when interrupt request signal is used as an event signal by the ELC. The event signal is output by the counter underflow and refresh error. An event signal is output regardless of the setting of the Reset Interrupt Request Select bit (WDTRCR.RSTIRQS) in register start mode or auto start mode. An event signal can also be output when the next interrupt source is generated while the Refresh Error flag (WDTSR.REFEF) or Underflow flag (WDTSR.UNDF) is 1. For details, see [section 18, Event Link Controller \(ELC\)](#).

## 25.5 Usage Notes

### 25.5.1 ICU Event Link Setting Register n (IELSRn) Setting

Setting 0x53 to ICU Event Link Setting Register n (ICU.IELSRn) is prohibited when WDT reset interrupt request selection resets (OFS0.WDTRSTIRQS = 0 or WDTRCR.RSTIRQS = 0) or when enabling event link operation (ELSRn.ELS[8:0] = 0x53).

## 26. Independent Watchdog Timer (IWDT)

### 26.1 Overview

The Independent Watchdog Timer (IWDT) consists of a 14-bit down counter that must be serviced periodically to prevent counter underflow. The IWDT provides functionality to reset the MCU or to generate a non-maskable interrupt or an underflow interrupt. Because the timer operates with an independent clock from CPU clock (CPUCLK) and System clock (ICLK), it is particularly useful in returning the MCU to a known state as a fail-safe mechanism when the system runs out of control. The IWDT can be triggered automatically by a reset, underflow, refresh error, or a refresh of the count value in the registers.

The IWDT functions differ from those of the WDT in the following respects:

The divided IWDT clock (IWDTCLK) is used as the count source (not affected by PCLKB)

Table 26.1 lists the IWDT specifications.

**Table 26.1 IWDT specifications**

Parameter	Specifications
Count source*1	IWDT clock (IWDTCLK)
Clock division ratio	Division by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	<ul style="list-style-type: none"> <li>Auto start mode: Counting automatically starts after a reset or after an underflow or refresh error occurs.</li> <li>Register start mode: Counting is started with a refresh by writing to the IWDTRR register.</li> <li>Only secure developer can select Auto-start mode or Register-start mode.</li> </ul>
Conditions for stopping the counter	<ul style="list-style-type: none"> <li>Reset (the down-counter and other registers return to their initial values).</li> <li>A counter underflows or a refresh error is generated.</li> </ul>
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods).
Independent watchdog timer reset sources	<ul style="list-style-type: none"> <li>Down-counter underflows</li> <li>Refreshing outside the refresh-permitted period (refresh error)</li> </ul>
Non-maskable interrupt/interrupt sources	<ul style="list-style-type: none"> <li>Down-counter underflows</li> <li>Refreshing outside the refresh-permitted period (refresh error)</li> </ul>
Reading of the counter value	The down-counter value can be read by the IWDTSR register.
Event link function (output)	<ul style="list-style-type: none"> <li>Down-counter underflow event output</li> <li>Refresh error event output</li> </ul>
Output signal (internal signal)	<ul style="list-style-type: none"> <li>Reset output</li> <li>Interrupt request output</li> <li>CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode and Deep Software Standby mode count stop control output.</li> </ul>
Trust Zone Filter	Security and Privilege attribution can be set.

Note 1. Satisfy the frequency of the peripheral module clock (PCLKB)  $\geq 4 \times$  (the frequency of the count clock source after division).  
The bus interface and registers operate with PCLKB, and the 14-bit counter and control circuits operate with IWDTCLK.

#### 26.1.1 Block diagram

In addition to the peripheral clock (PCLKB), an MCU needs the IWDT clock (IWDTCLK) that does not stop even in low power modes so that the MCU runs even in low power modes in which the peripheral clock (PCLKB) stops. The bus interface and registers operate with the peripheral clock (PCLKB), and the 14-bit down-counter and control circuits operate with the IWDT clock (IWDTCLK).

Signals are connected between the block operating with the peripheral clock (PCLKB) and the block operating with the IWDT clock (IWDTCLK) via a synchronization circuit.

Figure 26.1 shows a block diagram.

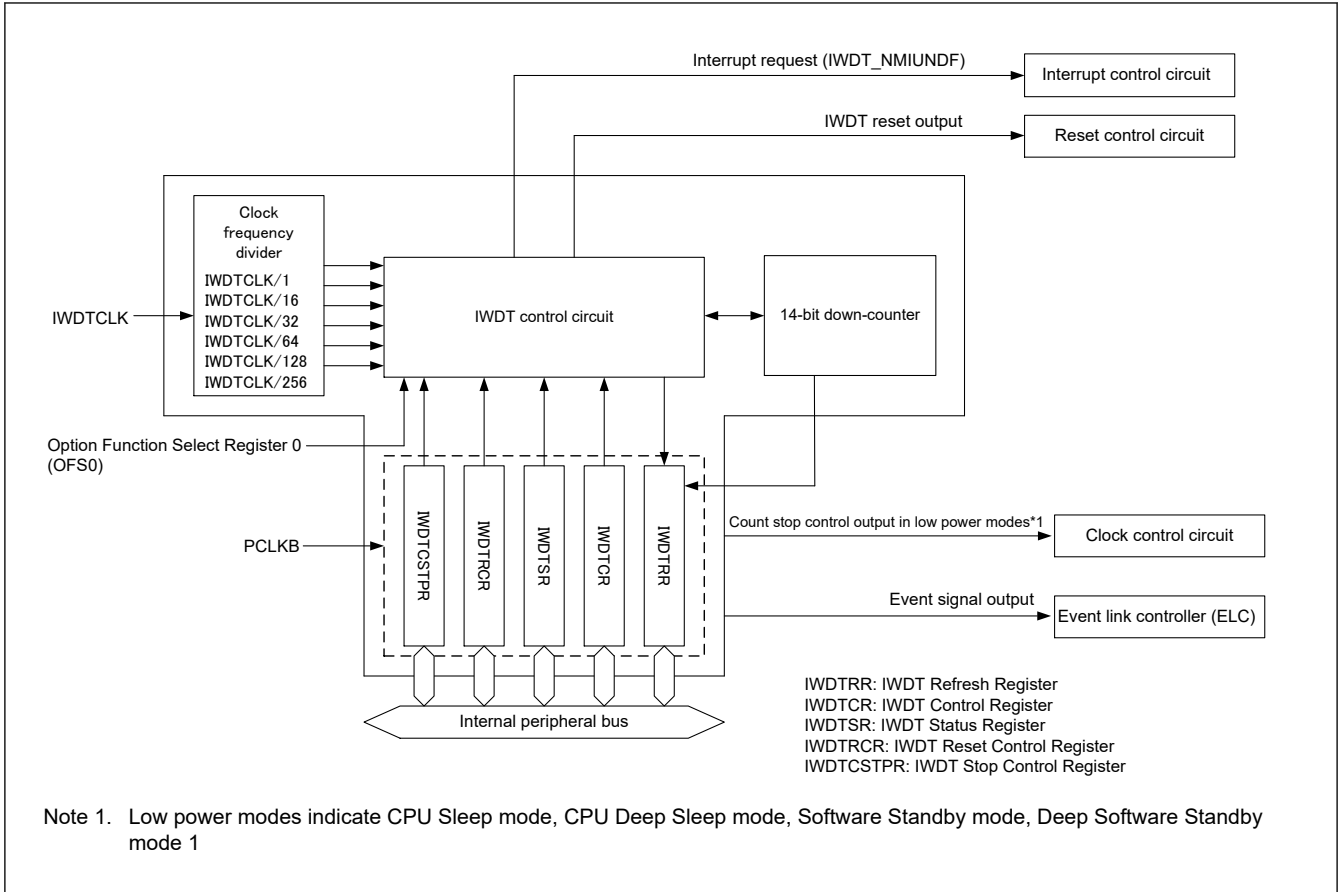


Figure 26.1 IWDT block diagram

## 26.2 Register specifications

### 26.2.1 IWDTRR : IWDT Refresh Register

Base address: IWDT = 0x4020\_2200  
 IWDT\_NS = 0x5020\_2200

Offset address: 0x0

Bit position: 7 6 5 4 3 2 1 0

Bit field: REFRESH[7:0]

Value after reset: 1 1 1 1 1 1 1 1

Bit	Symbol	Function	R/W
7:0	REFRESH[7:0]	Refresh Register The counter is refreshed by writing 0x00 and then writing 0xFF to this register.	R/W

Note: S-TYPE-3, P-TYPE-3

The IWDTRR register refreshes the down-counter of the IWDT.

#### REFRESH[7:0] bits (Refresh Register)

The down-counter of the IWDT is refreshed by writing 0x00 and then writing 0xFF to IWDTRR(refresh operation) within the refresh-permitted period. After the down-counter is refreshed, it starts counting down from the value selected by the IWDT time-out period selection bits (OFS0.IWDTTOPS[1:0]) in the auto start mode.

In the register start mode, counting down starts from the value set by the TOPS[1:0] bits in the IWDT Control Register (IWDTCR).

Also in the register start mode, counting down starts from the value set by the IWDTCR.TOPS[1:0] bits by the first refresh operation after the release from the reset state.

When 0x00 is written, the read value is always 0x00. When a value other than 0x00 is written, the read value is always 0xFF.

For details of the refresh operation, see [section 26.3.3. Refresh operation](#).

## 26.2.2 IWDTCR : IWDT Control Register

Base address: IWDT = 0x4020\_2200  
IWDT\_NS = 0x5020\_2200

Offset address: 0x2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit field:	—	—	RPSS[1:0]	—	—	RPES[1:0]	CKS[3:0]			—	—	TOPS[1:0]					
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Function	R/W
1:0	TOPS[1:0]	Timeout Period Select 0 0: 128 cycles (0x007F) 0 1: 512 cycles (0x01FF) 1 0: 1024 cycles (0x03FF) 1 1: 2048 cycles (0x07FF) Each value in parentheses () denotes a start value of down-counting.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	CKS[3:0]	Clock Division Ratio Select 0 0 0 0: IWDTCLK 0 0 1 0: IWDTCLK/16 0 0 1 1: IWDTCLK/32 0 1 0 0: IWDTCLK/64 1 1 1 1: IWDTCLK/128 0 1 0 1: IWDTCLK/256 Other settings are prohibited.	R/W
9:8	RPES[1:0]	Window End Position Select 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	RPSS[1:0]	Window Start Position Select 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The IWDTCR register is used to specify the timeout period (time until the down-counter underflows in the register start mode), clock division ratio, and the window start and end positions of refresh-permitted period. Some restrictions apply to writes to this register. For details, see [section 26.3.2. Controlling writes to the IWDTCR, IWDTRCR, and IWDTCSR registers](#).

In auto start mode, the settings of this register are disabled and the settings for the option function select register 0 (OFS0) are enabled. The same settings as in the bits in this register can be performed at the option function select register 0 (OFS0). For details, see [section 26.3.8. Correspondence between Option Function Select Register 0 \(OFS0\) and IWDT Registers](#).

### TOPS[1:0] bits (Timeout Period Select)

The TOPS[1:0] bits select the timeout period, the period until the down-counter underflows, from 128, 512, 1024, and 2048 cycles, taking the divided clock specified in the CKS[3:0] bits as one cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDT clock (IWDTCLK) cycles) until the counter underflows.

[section 26.2.2. IWDTCR : IWDT Control Register](#) shows the relationship among the CKS[3:0] and TOPS[1:0] bits settings, the timeout period, and the number of IWDT clock (IWDTCLK) cycles.

**Table 26.2 IWDT timeout period settings**

CKS[3:0]				TOPS[1:0]		Clock division ratio	Timeout period (Number of cycles)	Number of IWDT clock (IWDTCLK) cycles
0	0	0	0	0	0	IWDTCLK	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	IWDTCLK/16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	IWDTCLK/32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	IWDTCLK/64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	IWDTCLK/128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	IWDTCLK/256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

### CKS[3:0] bits (Clock Division Ratio Select)

These bits select IWDT clock (IWDTCLK) division ratio from among division by 1, 16, 32, 64, 128, and 256.

Combined with the TOPS[1:0] bits setting, a count period between 128 and 524288 cycles of the IWDT clock (IWDTCLK) can be selected for the IWDT.

Note: In order to read the down-counter value correctly, the relationship between the peripheral clock (PCLKB) frequency and the count clock (IWDTCLK) frequency must be set appropriately. For the setting, see Note1 in [Table 26.1](#)

### RPES[1:0] bits (Window End Position Select)

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the down-counter. The selected window end position should be a value smaller than the value for the window start position (that is, window start position > window end position). If the value for the window end position is greater than the value for the window start position, only the value for the window start position is effective.

The counter values for the window start and end positions selected by setting the RPES[1:0] and RPSS[1:0] bits change depending on the TOPS[1:0] bits setting.

[Table 26.3](#) lists the counter values for the window start and end positions corresponding to TOPS[1:0] bits values.

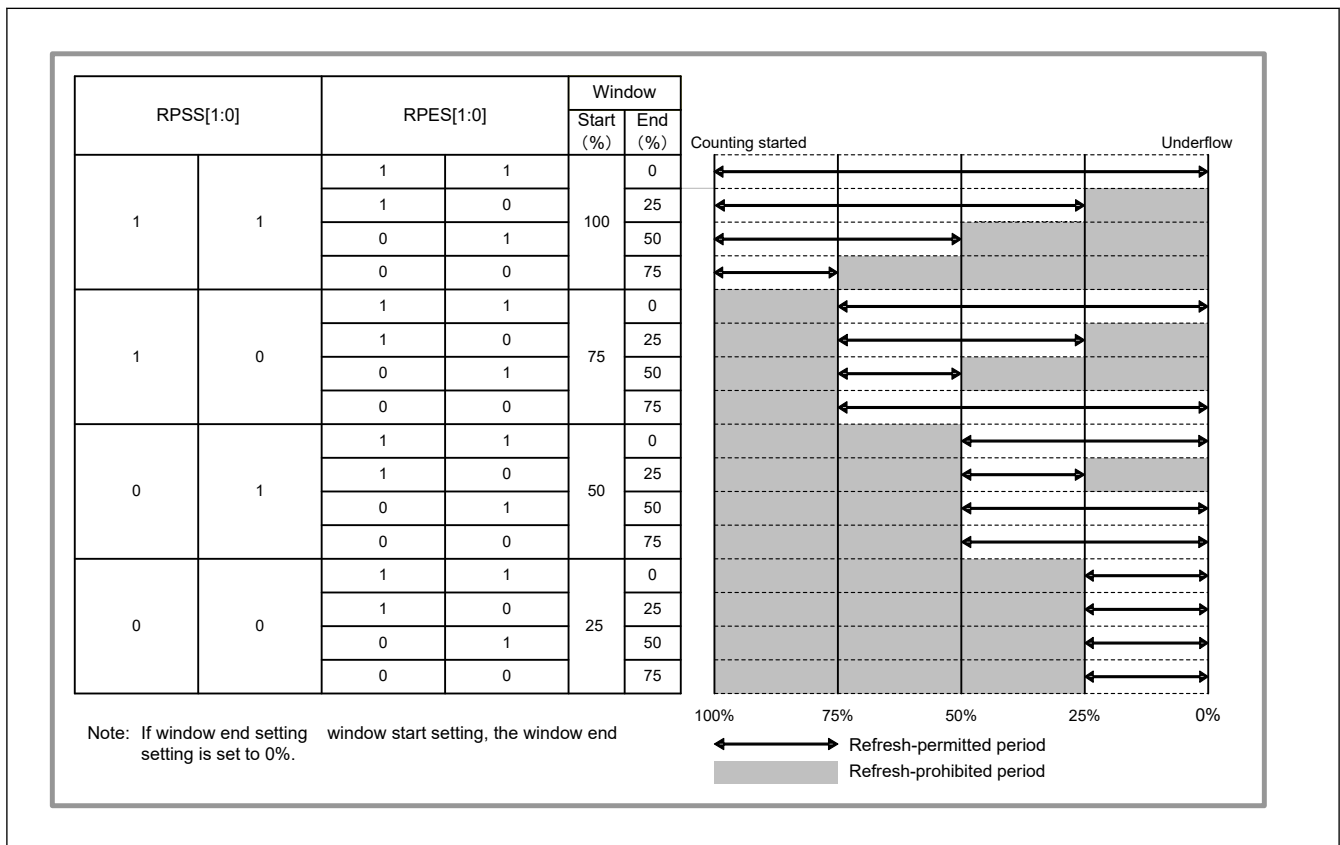
**Table 26.3 Relationship between timeout periods and window start and end counter values**

TOPS[1:0]		Timeout period		Window start and end counter value			
		Number of cycles	Counter value	100%	75%	50%	25%
0	0	128	0x007F	0x007F	0x005F	0x003F	0x001F
0	1	512	0x01FF	0x01FF	0x017F	0x00FF	0x007F
1	0	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
1	1	2048	0x07FF	0x07FF	0x05FF	0x03FF	0x01FF

**RPSS[1:0] bits (Window Start Position Select)**

These bits select a counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.

Figure 26.2 shows the relationship between the RPSS[1:0] and RPES[1:0] bits settings and the refresh-permitted and refresh-prohibited periods.



**Figure 26.2 RPSS[1:0] and RPES[1:0] bits settings and refresh-permitted and refresh-prohibited periods**

**26.2.3 IWDTSR : IWDT Status Register**

Base address: IWDT = 0x4020\_2200  
 IWDT\_NS = 0x5020\_2200

Offset address: 0x4

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	REFE F	UNDF F	CNTVAL[13:0]												
------------	-----------	-----------	--------------	--	--	--	--	--	--	--	--	--	--	--	--

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-Counter Value Counter value of the down-counter	R
14	UNDFE	Underflow Flag 0: No underflow occurred. 1: Underflow occurred.	R/W
15	REFEF	Refresh Error Flag 0: No refresh error occurred. 1: Refresh error occurred.	R/W

Note: S-TYPE-3, P-TYPE-3

The IWDTSR register indicates the counter value of the down-counter and whether an underflow or refresh error occurred in the down-counter.

It is initialized by the reset source of the IWDT. IWDTSR is not initialized by other reset sources.

### CNTVAL[13:0] bits (Down-Counter Value)

Read these bits to confirm the counter value of the down-counter.

Note: The read value might differ from the actual count by 1.

### UNDFE flag (Underflow Flag)

Read the UNDFE flag to confirm whether an underflow occurred in the down-counter.

A value of 1 indicates that the down-counter underflowed. A value of 0 indicates that the down-counter has not underflowed.

Write 0 to the UNDFE flag to clear the value. Writing 1 has no effect.

### REFEF flag (Refresh Error Flag)

Read the REFEF flag to confirm whether a refresh error (a refresh operation was performed during the prohibited period) occurred.

A value of 1 indicates that a refresh error occurred. A value of 0 indicates that no refresh error has occurred.

Write 0 to the REFEF flag to clear the value. Writing 1 has no effect.

## 26.2.4 IWDTRCR : IWDT Reset Control Register

Base address: IWDT = 0x4020\_2200  
IWDT\_NS = 0x5020\_2200

Offset address: 0x6

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RSTIR QS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	RSTIRQS	Reset Interrupt Request Select 0: Enable non-maskable interrupt requests or interrupt request output. 1: Enable reset output.	R/W

Note: S-TYPE-3, P-TYPE-3

IWDTRCR register controls the reset output or interrupt request output when the down-counter of the IWDT underflows. Some restrictions apply to writes to this register. For details, see [section 26.3.2. Controlling writes to the IWDTRCR, IWDTRCR, and IWDTCSR registers](#).

In auto start mode, the settings of this register are disabled and the settings for the input ports (OFS0) are enabled. The same settings as in the bits in this register can be performed at the input ports (OFS0). For details, see [section 26.3.8. Correspondence between Option Function Select Register 0 \(OFS0\) and IWDT Registers](#).



**RSTIRQS bit (Reset Interrupt Request Select)**

The RSTIRQS bit selects the reset output or interrupt request output when the down-counter underflows or a refresh error occurs.

**26.2.5 IWDTCSSTPR : IWDT Count Stop Control Register**

Base address: IWDT = 0x4020\_2200  
IWDT\_NS = 0x5020\_2200

Offset address: 0x8

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SLCS TP	—	—	—	—	—	—	—
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	—	This bit is read as 1. The write value should be 1.	R/W
7	SLCSTP	CPU Sleep-Mode Count Stop Select 0: Disable count stop. 1: Stop the counter when the CPU enters CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode, or Deep Software Standby mode 1.	R/W

Note: S-TYPE-3, P-TYPE-3

The IWDTCSSTPR register is used to stop the down-counter of the IWDT in case the CPU enters low-power modes. Some restrictions apply to writes to this register. For details, see [section 26.3.2. Controlling writes to the IWDTCSR, IWDTRCR, and IWDTCSSTPR registers](#).

In auto start mode, the settings of this register are disabled and the settings for the option function select register 0 (OFS0) are enabled. The same settings as in the bits in this register can be performed at the option function select register 0 (OFS0). For details, see [section 26.3.8. Correspondence between Option Function Select Register 0 \(OFS0\) and IWDT Registers](#).

**SLCSTP bit (CPU Sleep-Mode Count Stop Select)**

The SLCSTP bit selects whether to stop the counter in case the CPU enters CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode, or Deep Software Standby mode 1.

**26.3 Operation****26.3.1 Count operation in each start mode**

The start mode of the IWDT is selected by setting the IWDT start mode selection (OFS0.IWDTSTRT) bit during a reset.

When the IWDT start mode selection (OFS0.IWDTSTRT) bit is 1 (register start mode), the IWDT Control Register (IWDTCSR), IWDT Reset Control Register (IWDTRCR), and IWDT Count Stop Control Register (IWDTCSSTPR) are enabled, and counting is started by writing to the IWDT Refresh Register (IWDTRR) to refresh the counter. When the IWDT start mode selection (OFS0.IWDTSTRT) bit is 0 (auto start mode), the settings of the option function select register 0 (OFS0) are enabled, and counting automatically starts after a reset.

**(1) Register start mode**

When the register start mode is selected by setting the IWDT start mode selection (OFS0.IWDTSTRT) bit to 1 during a reset, the IWDT Control Register (IWDTCSR), IWDT Reset Control Register (IWDTRCR), and IWDT Count Stop Control Register (IWDTCSSTPR) are enabled.

After the reset state is released, set the clock division ratio, window start and end positions, and timeout period in the IWDTCSR register, the reset output or interrupt request output in the IWDTRCR register, and the IWDT down-counter stop control at transitions to low power modes in the IWDTCSSTPR register. Then, refresh the down-counter to set the value specified by the IWDTCSR.TOPS[1:0] bits in the counter and start counting down.

After that, as long as the program continues normal operation and the counter is refreshed in the refresh permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The IWDT does not output a reset signal as long as this continues.

However, if the down counter underflows because the down-counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request (IWDT\_NMIUNDF).

The reset output or interrupt request output is selected by setting the IWDTRCR.RSTIRQS bit.

Figure 26.3 shows an example of operation under the following conditions:

- Register start mode (OFS0.IWDTSTRT = 1)
- Reset output is enabled (IWDTRCR.RSTIRQS = 1)
- The window start position is 75% (IWDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (IWDTCR.RPES[1:0] = 10b)

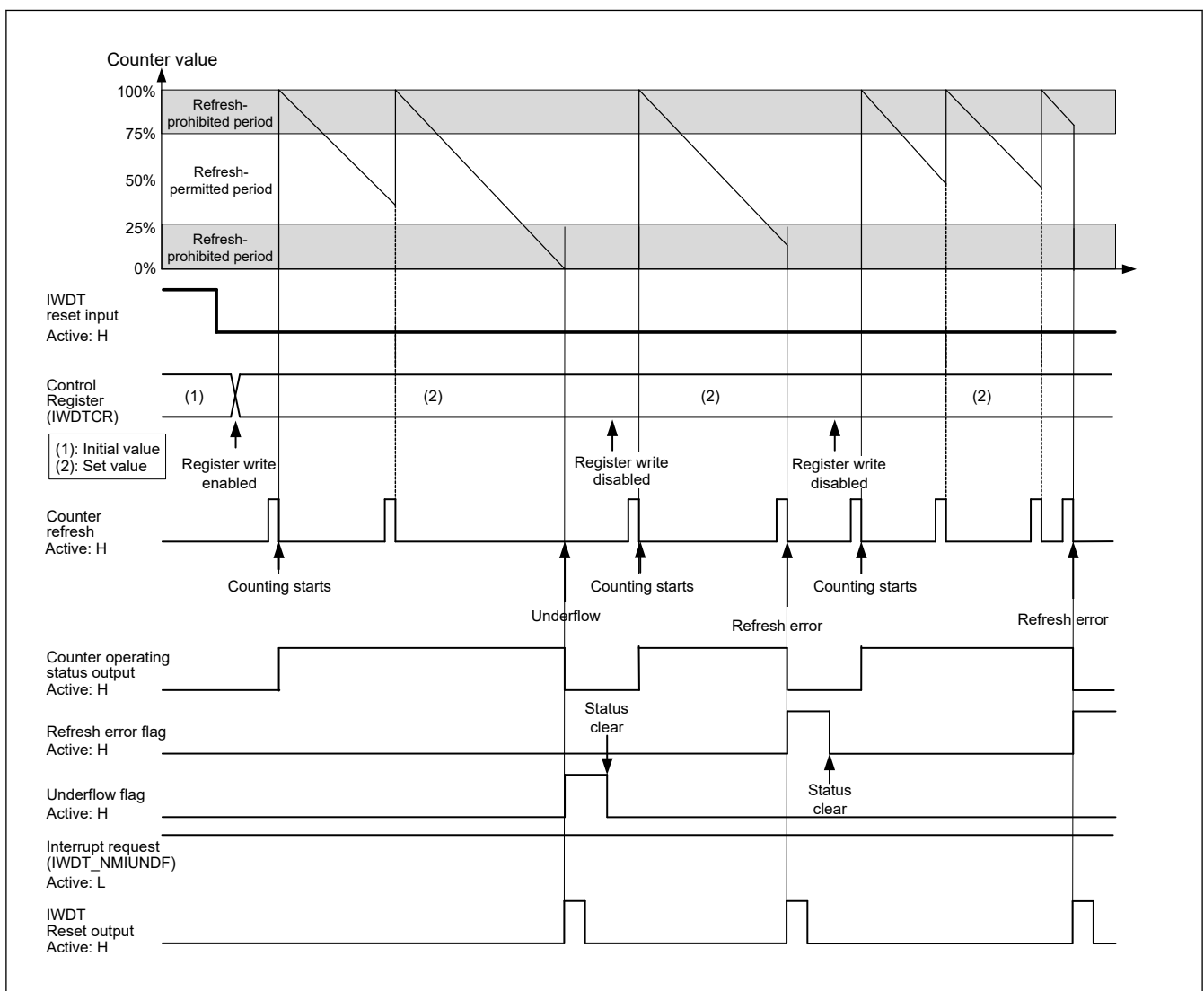


Figure 26.3 Operation example in register start mode

(2) Auto start mode

When the auto start mode is selected by setting the IWDT start mode selection (OFS0.IWDTSTRT) bit to 0 during a reset, the IWDT Control Register (IWDTCR), IWDT Reset Control Register (IWDTRCR), and IWDT Count Stop Control Register (IWDTCSTPR) are disabled.

During a reset, the clock division ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter stop control at transitions to low power modes should be set through the option function select register 0 (OFS0). Then, when the reset state is released, the timeout period set through the option function select register 0 (OFS0) is set in the down-counter, and then the down-counter automatically starts counting down.

After that, as long as the program continues normal operation and the counter is refreshed in the refresh permitted period, the value in the counter is reset each time the counter is refreshed and counting down continues. The IWDT does not output a reset signal as long as this continues.

However, if the down-counter underflows because the down counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request (IWDT\_NMIUNDF).

The reset output or interrupt request output is selected by setting the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS).

Figure 26.4 shows an example of operation under the following conditions:

- Auto start mode (OFS0.IWDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.IWDTRSTIRQS = 0)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDRPES[1:0] = 10b)

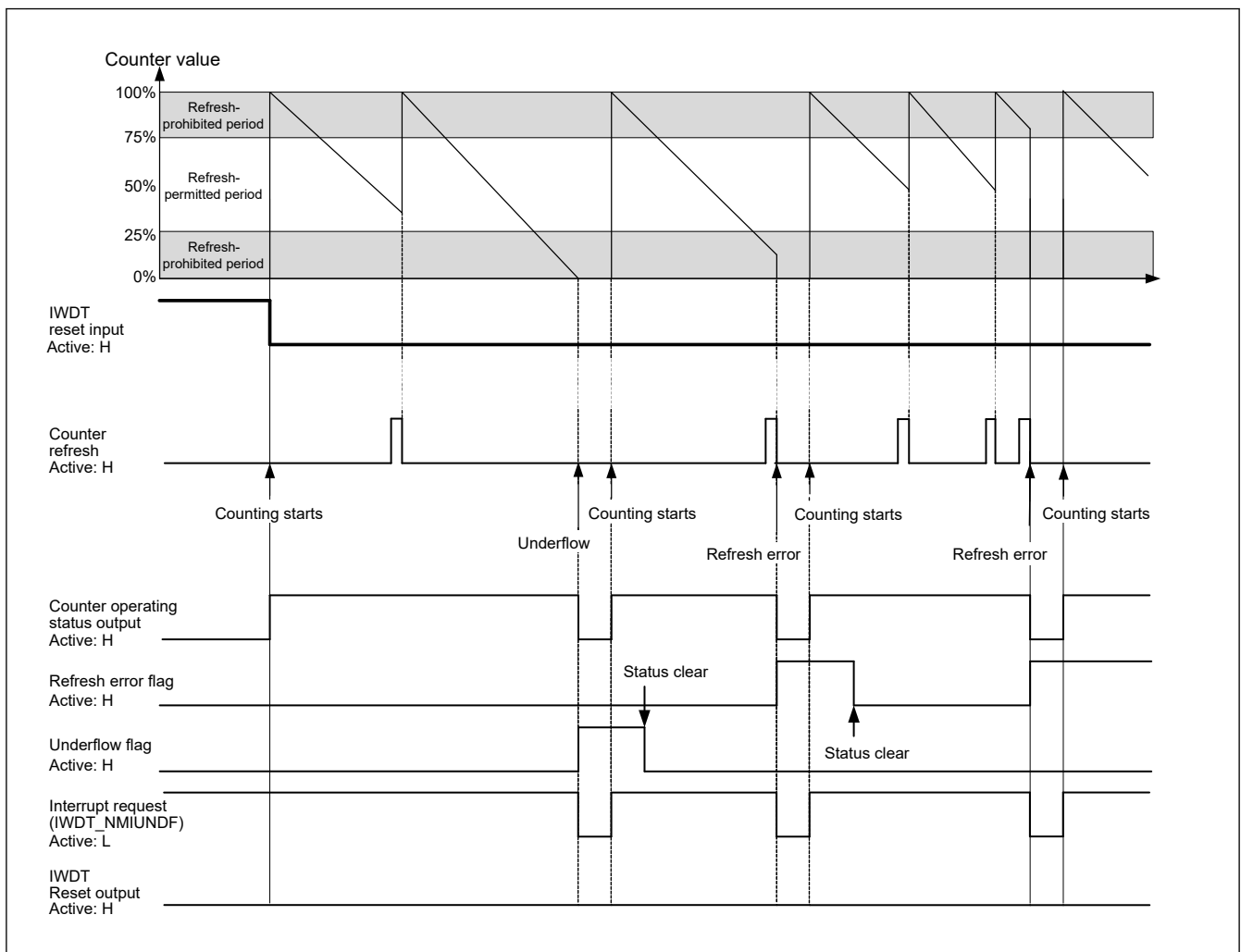


Figure 26.4 Operation example in auto start mode

### 26.3.2 Controlling writes to the IWDTCR, IWDTRCR, and IWDTCSSTPR registers

Writing to the IWDT Control Register (IWDTCR) is possible only once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or by writing to the IWDTCR register, the protection signal in the IWDT becomes 1 to protect the IWDTCR register against subsequent attempts of writing.

The IWDT Reset Control Register (IWDTRCR) and the IWDT Count Stop Control Register (IWDTCSSTPR) are also controlled in the same way.

This protection is released by a reset source to the IWDT. With other reset sources, the protection is not released.

Figure 26.5 shows control waveforms produced in response to writing to the IWDTCR register.

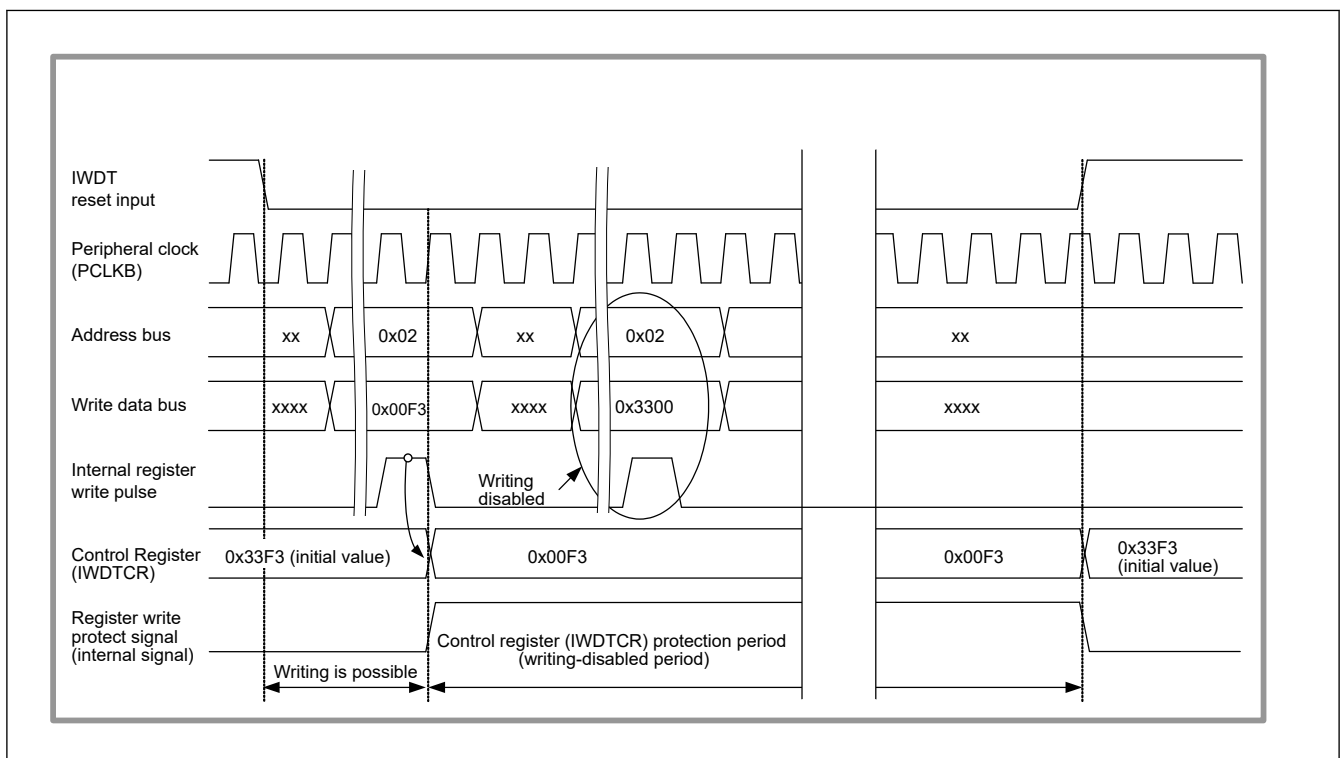


Figure 26.5 Control waveforms produced in response to writes to the IWDTCR register

### 26.3.3 Refresh operation

The down-counter is refreshed and starts operation (counting is started by refreshing) by writing the values 0x00 and then 0xFF to the IWDT Refresh Register (IWDTRR). If a value other than 0xFF is written after 0x00, the counter is not refreshed. After such invalid writing, correct refreshing is performed by writing 0x00 and then 0xFF again to the IWDT Refresh Register (IWDTRR).

When writing is done in the order of 0x00 (first time) → 0x00 (second time), and if 0xFF is written after that, the writing order 0x00 → 0xFF is satisfied; writing 0x00 ((n-1) th time) → 0x00 (nth time) → 0xFF is valid and correct refreshing will be done. Even when the first value written before 0x00 is not 0x00, correct refreshing will be done if the operation contains the sequence of writing 0x00 → 0xFF. Correct refreshing is also performed when a register other than IWDTRR is accessed or IWDTRR is read between writing 0x00 and writing 0xFF to IWDTRR.

[Example write sequences that are valid for refreshing the counter]

- 0x00 → 0xFF
- 0x00 ((n-1) th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from IWDTRR → 0xFF

[Example write sequences that are invalid for refreshing the counter]

- 0x23 (a value other than 0x00) → 0xFF

- 0x00 → 0x54 (a value other than 0xFF)
- 0x00 → 0xAA (0x00 and a value other than 0xFF) → 0xFF

Even when 0x00 is written to IWDTRR outside the refresh-permitted period, if 0xFF is written to IWDTRR in the refresh-permitted period, the writing sequence is valid and refreshing will be done.

After 0xFF is written to the IWDTRR register, refreshing the counter requires up to four cycles of the signal for counting (the clock division ratio select bits (IWDTCR.CKS[3:0]) determine how many cycles of the IWDT clock (IWDTCLK) make up one cycle for counting).

Therefore, writing 0xFF to the IWDTRR register should be completed four count cycles before the end position of the refresh-permitted period or the down-counter underflow. The value of the down-counter can be checked using the down-counter value bits (IWDTSR.CNTVAL[13:0]).

[Example refreshing timings]

- When the window start position is set to 0x1FFF, even if 0x00 is written to IWDTRR before 0x1FFF is reached (at 0x2002, for example), refreshing occurs if 0xFF is written to the IWDTRR register after the value of the IWDTSR.CNTVAL[13:0] bits reaches 0x1FFF.
- When the window end position is set to 0x1FFF, refreshing is done if 0x2003 (four count cycles before 0x1FFF) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 0x00 → 0xFF to the IWDTRR register.
- When the refresh-permitted period continues until count 0x0000, refreshing can be done immediately before an underflow. In this case, if 0x0003 (four count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 0x00 → 0xFF to the IWDTRR register, no underflow occurs and refreshing is done.

Figure 26.6 shows the IWDT refresh operation waveforms when the peripheral clock (PCLKB) > IWDT clock (IWDTCLK) and the clock division ratio = IWDTCLK/1. Figure 26.7 shows the IWDT refresh operation waveforms when the peripheral clock (PCLKB) < IWDT clock (IWDTCLK) and the clock division ratio = IWDTCLK/16.

When transitioning to Software Standby mode or Deep Software Standby mode 1, the refresh procedure must be completed before the transitioning (0xFF is written after 0x00 is written to the IWDTRR register).

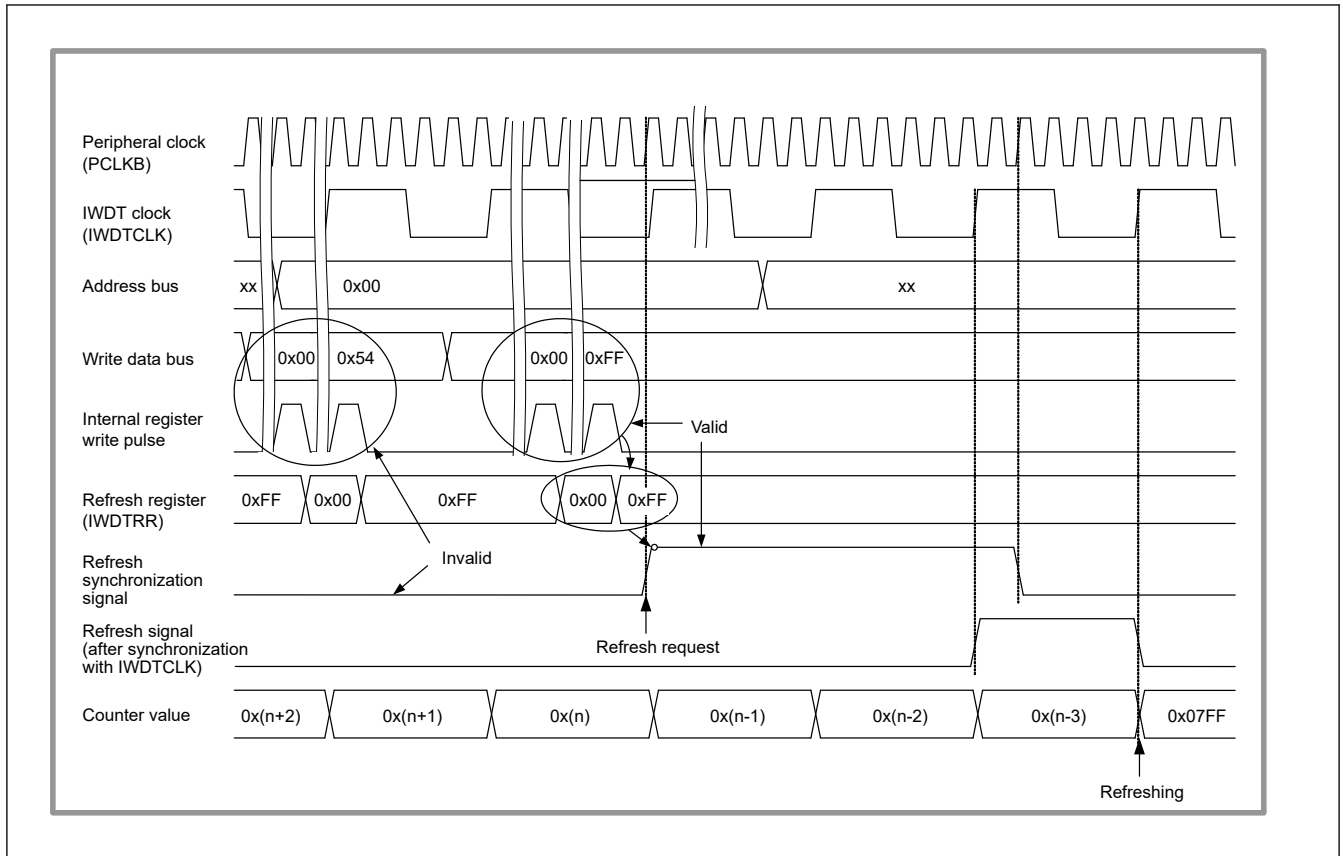


Figure 26.6 IWDT refresh operation waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

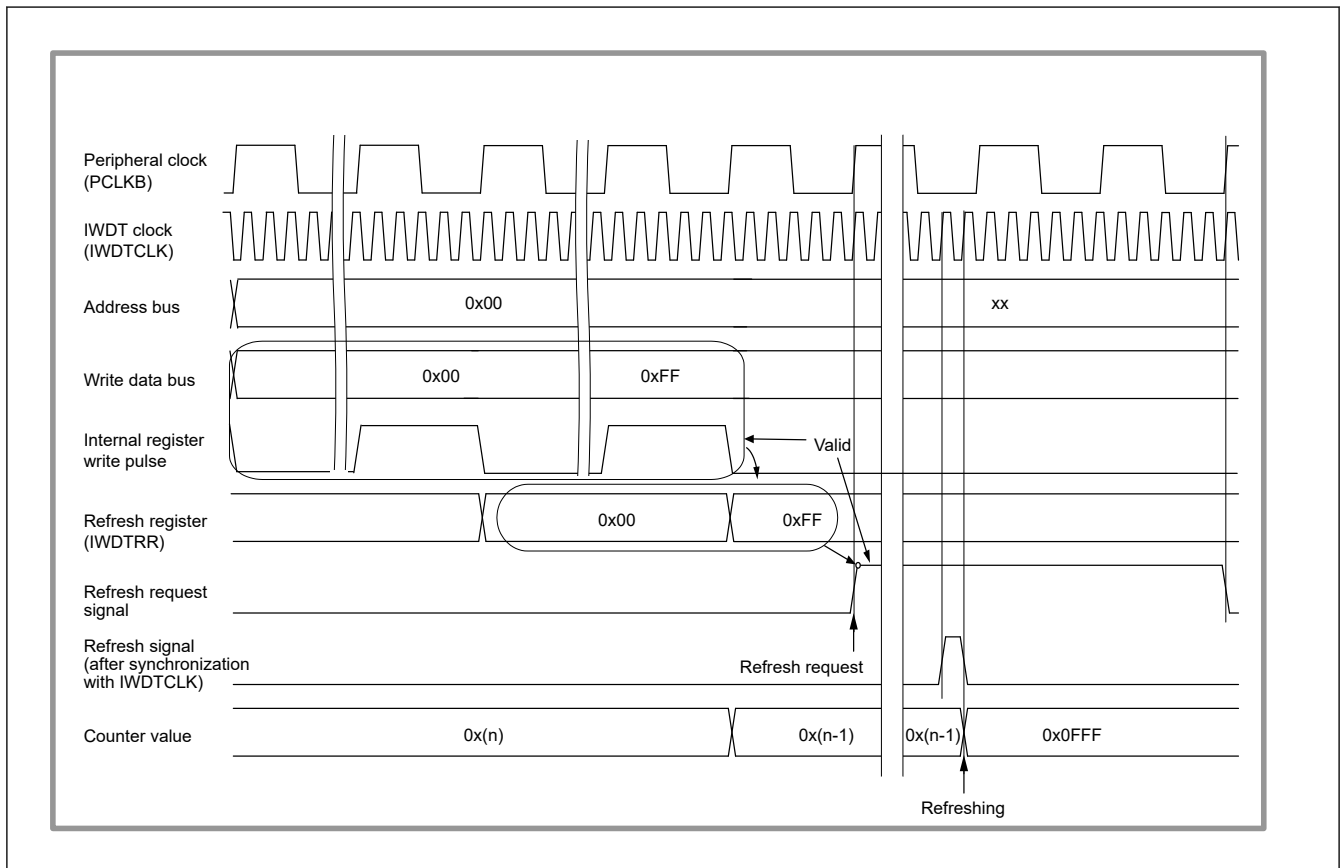


Figure 26.7 IWDT refresh operation waveforms (IWDTCR.CKS[3:0] = 0010b, IWDTCR.TOPS[1:0] = 01b)

### 26.3.4 Status flags

The refresh error flag (IWDTSR.REFEF) and underflow flag (IWDTSR.UNDF) retain the source of the reset output from the IWDT or the source of the interrupt request from the IWDT.

Thus, after release from the reset state or interrupt request generation, read the IWDTSR.REFEF and IWDTSR.UNDF flags to check for the reset or interrupt source.

For each flag, writing 0 clears the bit and writing 1 has no effect.

Leaving each flag unchanged does not affect operation. If the flags are not cleared, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written when the IWDT outputs a reset signal next.

### 26.3.5 Reset output

When the reset interrupt select bit (IWDTRCR.RSTIRQS) is set to 1 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) is set to 1 in auto start mode, a reset signal is output for one count cycle when the down counter overflows or a refresh error occurs. In register start mode, the down counter is initialized (all bits set to 0) and kept in that state after assertion of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing. In auto start mode, counting down automatically starts after the reset output.

### 26.3.6 Interrupt sources

When the reset interrupt select bit (IWDTRCR.RSTIRQS) is set to 0 in register start mode or when the IWDT reset interrupt request select bit (OFS0.IWDTRSTIRQS) is set to 0 in auto start mode, a non-maskable interrupt (IWDT\_NMIUNDF) is generated for one-count cycle when the down-counter overflows or a refresh error occurs.

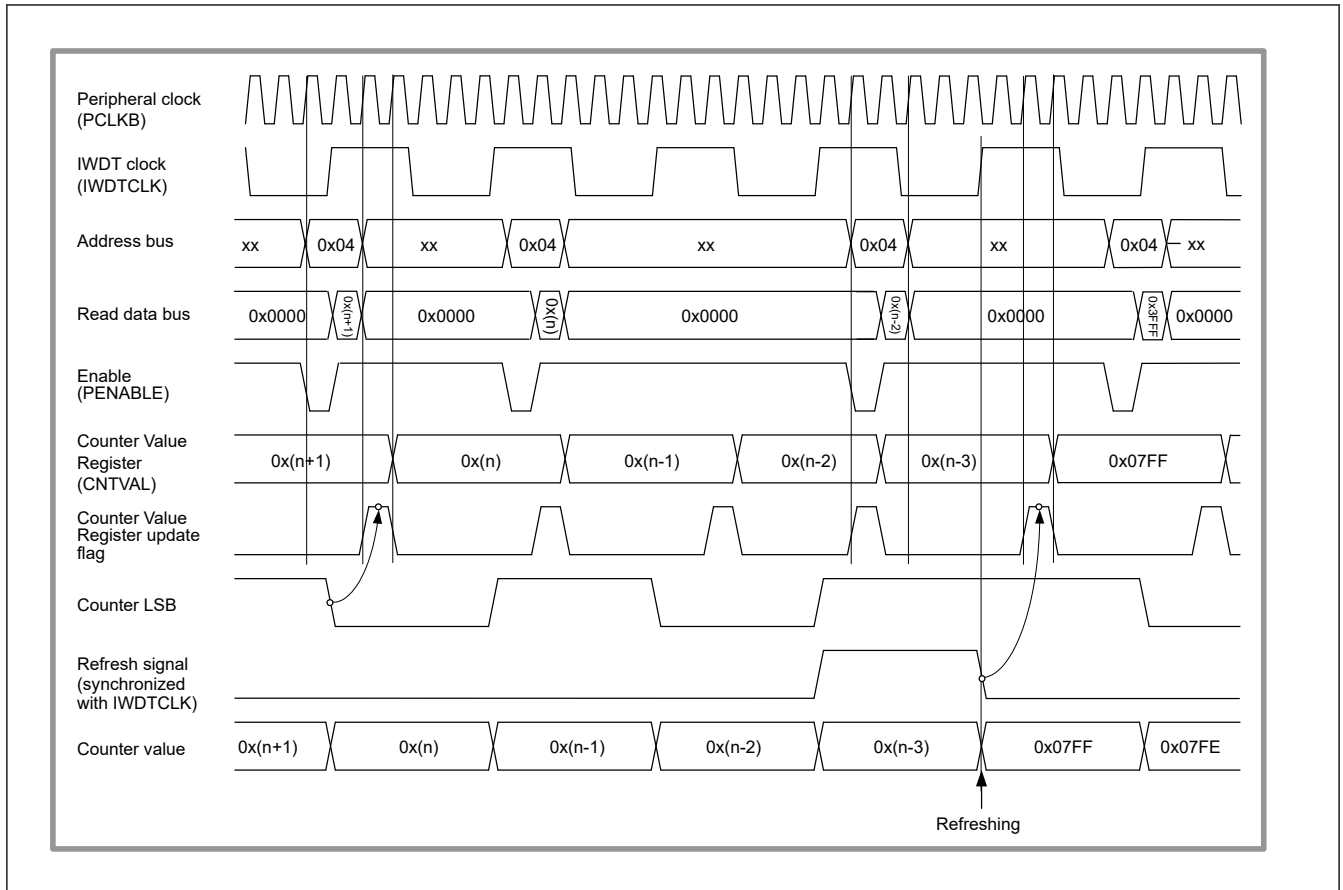
### 26.3.7 Reading the down-counter value

Because the down-counter in the IWDT operates on the IWDT clock (IWDTCLK), the counter value cannot be read directly.

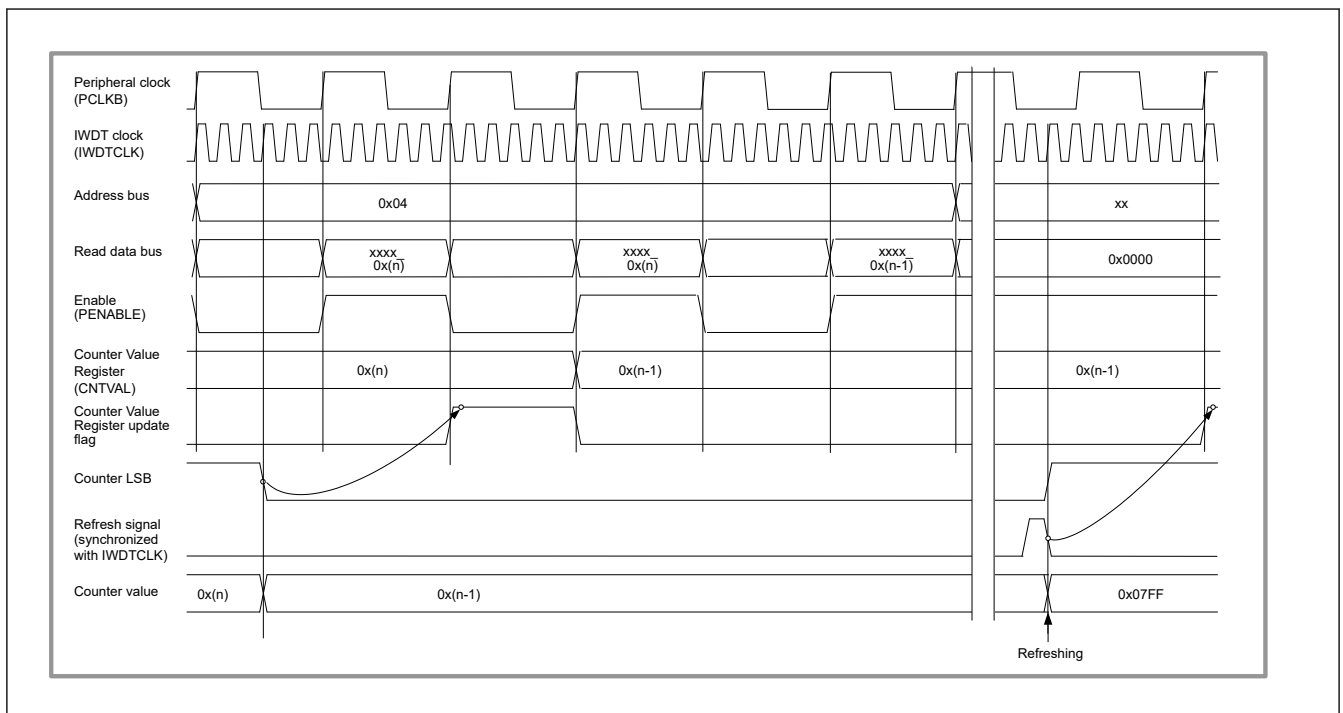
Therefore, the IWDT synchronizes the counter value with the peripheral clock (PCLKB) and stores it in the down-counter value bits (CNTVAL[13:0]) of the IWDT Status Register (IWDTSR). The counter value can be checked indirectly by reading the value stored in the IWDTSR.CNTVAL[13:0] bits.

Reading the counter value requires multiple peripheral clock (PCLKB) cycles (up to four clock cycles), and the read counter value may differ from the actual counter value by a value of one count.

Figure 26.8 shows the processing for reading the IWDT counter value when the peripheral clock (PCLKB) > IWDT clock (IWDTCLK) and the clock division ratio = IWDTCLK/1. Figure 26.9 shows the processing for reading the IWDT counter value when the peripheral clock (PCLKB) < IWDT clock (IWDTCLK) and the clock division ratio = IWDTCLK/16.



**Figure 26.8 Processing for reading IWDT counter value when IWDTCR.CKS[3:0] = 0000b and IWDTCR.TOPS[1:0] = 11b**



**Figure 26.9 Processing for reading IWDT counter value when IWDTCR.CKS[3:0] = 0010b and IWDTCR.TOPS[1:0] = 11b**



### 26.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 26.4 lists the correspondence between the option function select register 0 (OFS0) and the IWDT registers (IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR)) regarding control of the down-counter, reset or interrupt request output, and count stop function.

Control can be switched between the option function select register 0 (OFS0) and the IWDT registers (IWDTCR, IWDTRCR, and IWDTCSTPR) through the setting of the IWDT start mode selection (OFS0.IWDTSTRT) bit.

Note that the option function select register 0 (OFS0) setting should be kept unchanged during IWDT operation.

**Table 26.4 Association between Option Function Select Register 0 (OFS0) and the IWDT register**

Control	Function	OFS0 register (enabled in auto start mode) OFS0.IWDTSTRT = 0	IWDT register (enabled in register start mode) OFS0.IWDTSTRT = 1
Down-counter	Timeout period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	Window start position selection	OFS0.IWDRPSS[1:0]	IWDTCR.RPSS[1:0]
	Window end position selection	OFS0.IWDRPES[1:0]	IWDTCR.RPES[1:0]
Reset output/ interrupt request output	Reset output/ interrupt request output select	OFS0.IWDRSTIRQS	IWDTRCR.RSTIRQS
Stop the counter	CPU Sleep-mode count stop selection	OFS0.IWDTSLCSTP	IWDTCSTPR.SLCSTP

## 26.4 Link operation by the event link function

The IWDT is capable of link operation for a specified module when an interrupt request signal is used as an event signal. The IWDT outputs an event signal on an underflow of the down-counter or generation of a refresh error.

It outputs an event signal regardless of the setting of the reset interrupt request select bit (IWDTRCR.RSTIRQS) in register start mode or the IWDT reset interrupt request select bit (OFS0.IWDRSTIRQS) in auto start mode. It can output an event signal at generation of the next interrupt source while retaining the refresh error flag (IWDTSR.REFEF) or underflow flag (IWDTSR.UNDFE).

## 27. Ethernet MAC Controller (ETHERC)

### 27.1 Overview

The MCU provides a one-channel Ethernet Controller (ETHERC) compliant with the Ethernet or IEEE802.3 Media Access Control (MAC) layer protocol. Each ETHERC channel has one channel of the MAC layer interface. Connecting the MCU to the physical layer LSI (PHY-LSI) allows transmission and reception of frames compliant with the Ethernet/IEEE802.3 standard. The ETHERC is connected to the Ethernet DMA Controller (EDMAC), so data can be transferred without using the CPU.

Table 27.1 lists the ETHERC specifications, Figure 27.1 shows the configuration, and Table 27.2 lists the I/O pins.

Figure 27.1 and Figure 27.3 show examples connections of the MCU to an external PHY-LSI.

**Table 27.1 ETHERC specifications**

Parameter	Specifications
Number of channels	One channel
Protocol	Flow control compliant with IEEE802.3x
Data transmission/reception	Frames compliant with the Ethernet/IEEE802.3 standard can be transmitted and received
Bit rate	Supports 10 Mbps and 100 Mbps
Operation modes	Supports full-duplex and half-duplex modes
Interfaces	Media Independent Interface (MII), Reduced Media Independent Interface (RMII), compliant with the IEEE802.3u standard
Functions	<ul style="list-style-type: none"> <li>• Magic Packet™ detection</li> <li>• Wake-on-LAN (WOL) signal output</li> </ul>
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set.

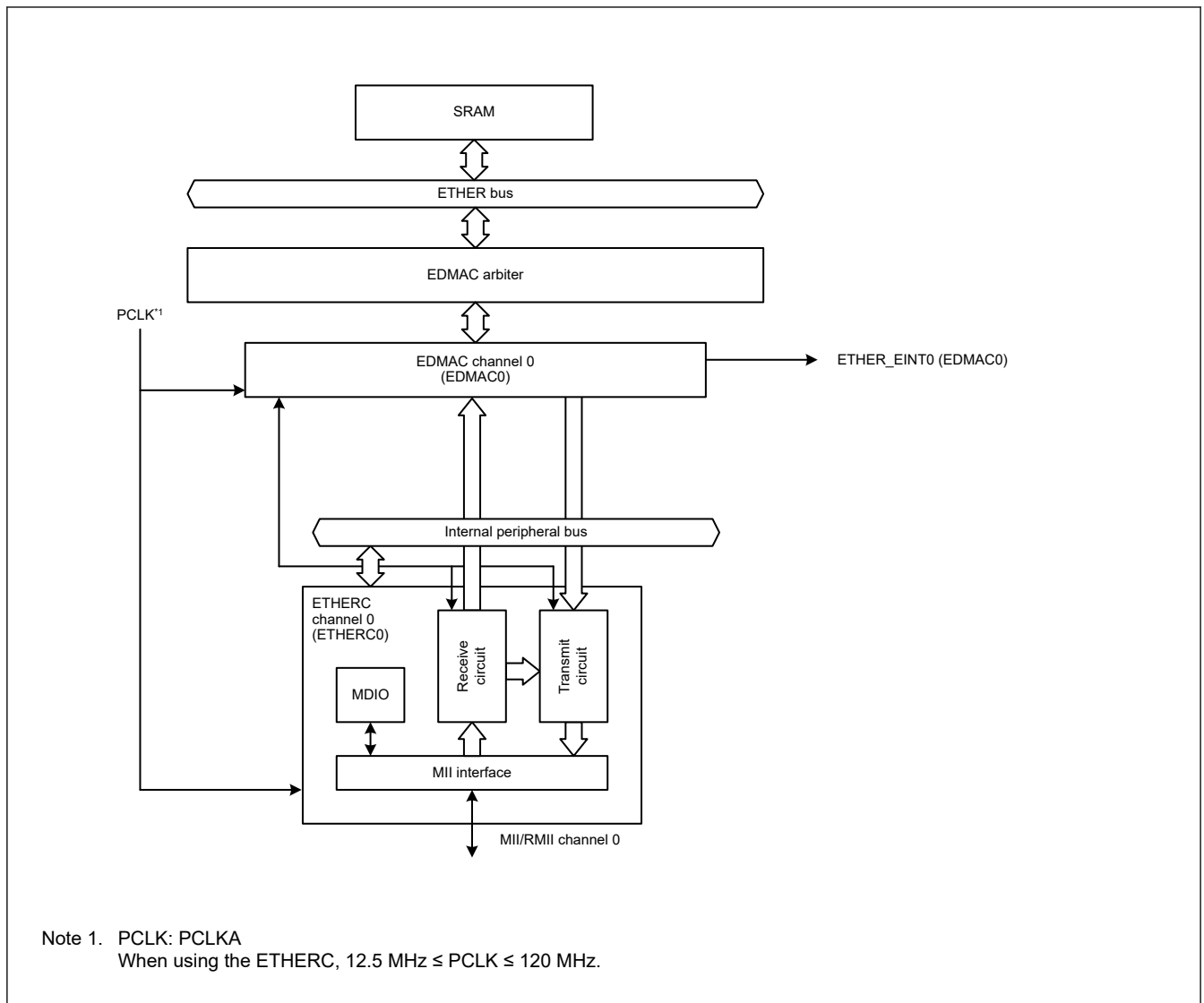


Figure 27.1 ETHERC configuration

Table 27.2 ETHERC I/O pins (1 of 2)

Operating mode	Pin name	I/O	Description
MII	ET0_TX_CLK *1	Input	Transmit clock Timing reference signal for outputting the ET0_TX_EN, ET0_ETXD3 to ET0_ETXD0, and ET0_TX_ER signals.
	ET0_RX_CLK *1	Input	Receive clock Timing reference signal for inputting the ET0_RX_DV, ET0_ERXD3 to ET0_ERXD0, and ET0_RX_ER signals.
	ET0_TX_EN *1	Output	Transmit data valid This signal indicates that valid transmit data was output on pins ET0_ETXD3 to ET0_ETXD0.
	ET0_ETXD3 to ET0_ETXD0 *1	Output	4-bit transmit data
	ET0_TX_ER *1	Output	Transmit error This signal notifies the PHY-LSI that an error occurred during transmission.
	ET0_RX_DV *1	Input	Receive data valid This signal indicates that valid receive data is on pins ET0_ERXD3 to ET0_ERXD0.
	ET0_ERXD3 to ET0_ERXD0 *1	Input	4-bit receive data
	ET0_RX_ER *1	Input	Receive error This signal indicates that there is an error in a frame that is being transferred from the PHY-LSI to the ETHERC.
	ET0_CRS *1	Input	Carrier sense
	ET0_COL *1	Input	Collision detection signal
	ET0_MDC *1	Output	Management data clock Reference clock signal for transfer of information on the ET0_MDIO pin.
	ET0_MDIO *1	I/O	Management data Input/Output Bidirectional data signal for exchanging management data with the PHY-LSI.
	ET0_LINKSTA	Input	Link status input from the PHY-LSI
	ET0_EXOUT	Output	General output pin
ET0_WOL	Output	Wake-on-LAN. This signal indicates that a Magic Packet was received	

Table 27.2 ETHERC I/O pins (2 of 2)

Operating mode	Pin name	I/O	Description
RMII	REF50CK0 *2	Input	Reference clock Timing reference signal for the RMII0_TXD_EN, RMII0_TXD1 to RMII0_TXD0, RMII0_CRS_DV, RMII0_RXD1 to RMII0_RXD0, and RMII0_RX_ER pins.
	RMII0_TXD_EN *2	Output	Transmit data valid This signal indicates that valid transmit data was output on the RMII0_TXD1 and RMII0_TXD0 pins.
	RMII0_TXD1 to RMII0_TXD0 *2	Output	2-bit transmit data
	RMII0_CRS_DV *2	Input	Carrier sense/receive data valid This signal indicates that valid receive data is on the RMII0_RXD1 and RMII0_RXD0 pins.
	RMII0_RXD1 to RMII0_RXD0 *2	Input	2-bit receive data
	RMII0_RX_ER *2	Input	Receive error This signal indicates that there is an error in a frame that is being transferred from the PHY-LSI to the ETHERC. See the note in <a href="#">section 27.5.2. Input to RMII0_RX_ER Pin While RMII Is Selected.</a>
	ET0_MDC *2	Output	Management data clock Reference clock signal for transfer of information on the ET0_MDIO pin
	ET0_MDIO *2	I/O	Management data Input/Output Bidirectional data signal for exchanging management data with the PHY-LSI.
	ET0_LINKSTA	Input	Link status input from the PHY-LSI.
	ET0_EXOUT	Output	General output pin
ET0_WOL	Output	Wake-on-LAN. This signal indicates that a Magic Packet was received.	

Note 1. MII signal compliant with IEEE802.3u.

Note 2. RMII signal compliant with IEEE802.3u.

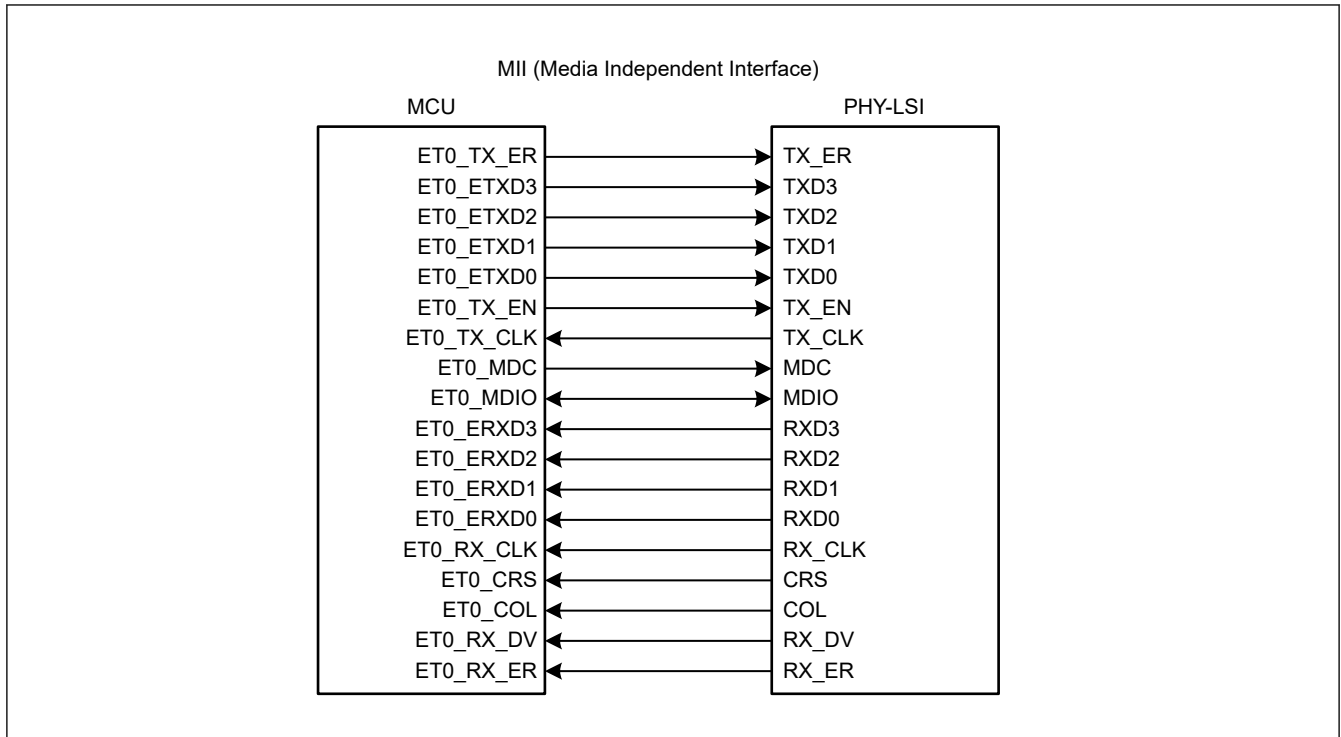


Figure 27.2 Example of connection with PHY-LSI for MII

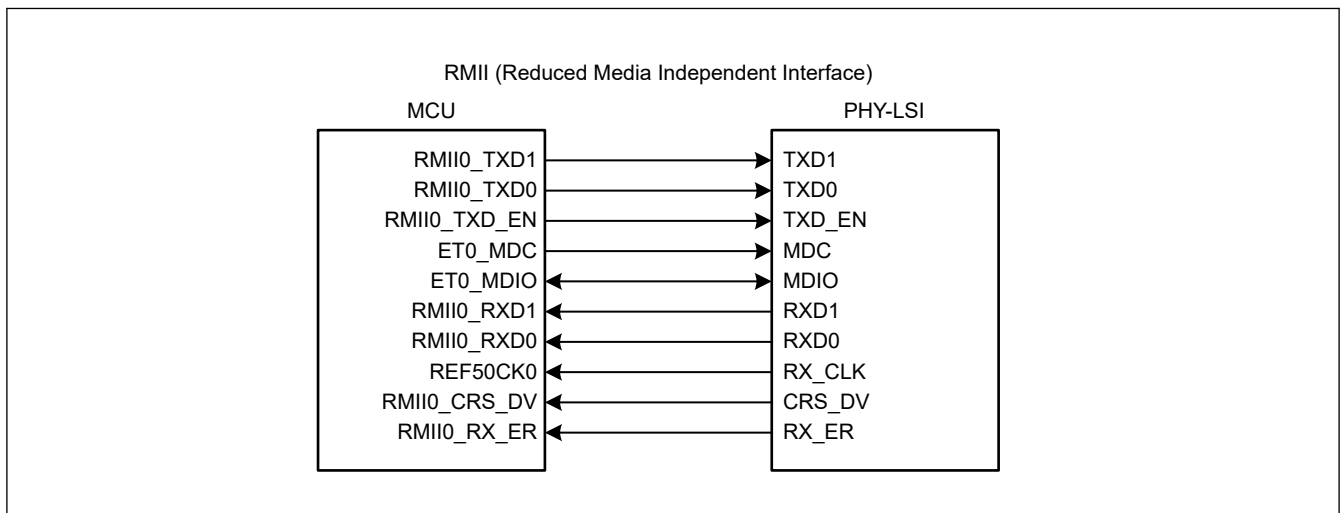


Figure 27.3 Example of connection with PHY-LSI for RMII

## 27.2 Register Descriptions

### 27.2.1 ECMR : ETHERC Mode Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TPC	ZPF	PFR	RXF	TXF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PRCE F	—	—	MPDE	—	—	RE	TE	—	ILB	RTM	DM	PRM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRM	Promiscuous Mode 0: Disable promiscuous mode 1: Enable promiscuous mode.	R/W
1	DM	Duplex Mode 0: Half-duplex mode 1: Full-duplex mode.	R/W
2	RTM	Bit Rate 0: 10 Mbps 1: 100 Mbps.	R/W
3	ILB	Internal Loopback Mode 0: Perform normal data transmission or reception 1: Loop data back in the ETHERC when full-duplex mode is selected.	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	TE	Transmission Enable 0: Disable transmit function 1: Enable transmit function.	R/W
6	RE	Reception Enable 0: Disable receive function 1: Enable receive function.	R/W
8:7	—	These bits are read as 0. The write value should be 0.	R/W
9	MPDE	Magic Packet Detection Enable 0: Disable Magic Packet detection 1: Enable Magic Packet detection.	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	PRCEF	CRC Error Frame Receive Mode 0: Notify EDMAC of a CRC error 1: Do not notify EDMAC of a CRC error.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	TXF	Transmit Flow Control Operating Mode 0: Disable automatic PAUSE frame transmission (PAUSE frame is not automatically transmitted) 1: Enable automatic PAUSE frame transmission (PAUSE frame is automatically transmitted as required).	R/W
17	RXF	Receive Flow Control Operating Mode 0: Disable PAUSE frame detection 1: Enable PAUSE frame detection.	R/W

Bit	Symbol	Function	R/W
18	PFR	PAUSE Frame Receive Mode 0: Do not transfer PAUSE frame to the EDMAC 1: Transfer PAUSE frame to the EDMAC.	R/W
19	ZPF	0 Time PAUSE Frame Enable 0: Do not use PAUSE frames that containing a pause_time parameter of 0 1: Use PAUSE frames that containing a pause_time parameter of 0.	R/W
20	TPC	PAUSE Frame Transmit 0: Transmit PAUSE frame even during a PAUSE period 1: Do not transmit PAUSE frame during a PAUSE period.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The ECMR register controls ETHERC operation. Except for the TE and RE bits, set the bits in this register during initialization after a reset. When rewriting this register outside the initialization process, set the EDMAC0.EDMR.SWR bit to 1 to reset the EDMAC and ETHERC, then set this register again.

#### PRM bit (Promiscuous Mode)

When the PRM bit is set to 1, the ETHERC operates in promiscuous mode, where all Ethernet frames are received. In promiscuous mode, the ETHERC receives all valid frames regardless of whether the address matches the destination or broadcast address, and regardless of the multicast bit setting.

#### RTM bit (Bit Rate)

The RTM bit sets the bit rate when the RMII is selected.

#### ILB bit (Internal Loopback Mode)

When the ILB bit is set to 1, transmit frames can be looped back in the MCU. Set the DM bit to 1 (full-duplex mode) to perform a loopback test.

#### TE bit (Transmission Enable)

When the TE bit is set to 1, the ETHERC transmit function is enabled. When the TE bit is set to 0, the transmit function is disabled after the frame being processed is completely transmitted.

#### RE bit (Reception Enable)

When the RE bit is set to 1, the ETHERC receive function is enabled. When the RE bit is set to 0, the receive function is disabled after the frame being processed is completely received.

#### PRCEF bit (CRC Error Frame Receive Mode)

When the PRCEF bit is set to 1, the EDMAC is not notified that a CRC error has occurred even when the error is detected in a receive frame. Accordingly, the EDMAC0.EESR.CERF flag and RFS0 bit in receive descriptor 0 (RD0) do not set to 1.

#### ZPF bit (0 Time PAUSE Frame Enable)

When the ZPF bit is 1, a PAUSE frame with a pause\_time parameter of 0 is transmitted when a PAUSE frame transmit request is canceled before the PAUSE time of the previously transmitted PAUSE frame has elapsed. After the PAUSE frame containing the pause\_time parameter of 0 is received, the ETHERC is ready for transmission.

When the ZPF bit is 0, even if the PAUSE frame transmit request from the receive FIFO is canceled, the next PAUSE frame is not transmitted until the PAUSE time of the previously transmitted PAUSE frame has elapsed. When a PAUSE frame containing a pause\_time parameter of 0 is received, it is discarded.

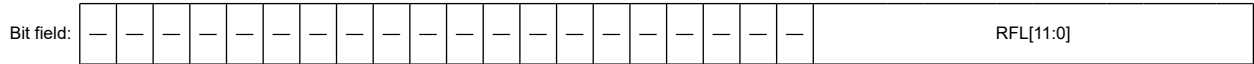


### 27.2.2 RFLR : Receive Frame Maximum Length Register

Base address: ETHERC0 = 0x4035\_4100  
 ETHERC0\_NS = 0x5035\_4100

Offset address: 0x08

Bit position: 31 11 0



Value after reset: 0

Bit	Symbol	Function	R/W
11:0	RFL[11:0]	Receive Frame Maximum Length The set value becomes the maximum frame length. The minimum value that can be set is 1,518 bytes, and the maximum value that can be set is 2,048 bytes. Values less than 1,518 bytes are regarded as 1,518 bytes, and values larger than 2,048 bytes are regarded as 2,048 bytes.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RFLR register specifies the maximum frame length that can be received by the MCU. Set the length in bytes. Do not rewrite this register while the ECMR.RE bit is 1 (receive function enabled).

#### RFL[11:0] bits (Receive Frame Maximum Length)

The RFL[11:0] bits set the frame length to be checked. The frame length is the number of bytes in a field, extending from the destination address to the frame check sequence [FCS] of the received frame. When this length exceeds the RFL[11:0] bit value, the EDMAC is notified of a frame-too-long error, and the excess data is discarded.

### 27.2.3 ECSR : ETHERC Status Register

Base address: ETHERC0 = 0x4035\_4100  
 ETHERC0\_NS = 0x5035\_4100

Offset address: 0x10

Bit position: 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	ICD	False Carrier Detect Flag 0: PHY-LSI has not detected a false carrier on the line 1: PHY-LSI detected a false carrier on the line.	R/W <sup>1</sup>
1	MPD	Magic Packet Detect Flag 0: Magic Packet not detected 1: Magic Packet detected.	R/W <sup>1</sup>
2	LCHNG	Link Signal Change Flag 0: Change in the ET0_LINKSTA signal not detected 1: Change in the ET0_LINKSTA signal detected (high to low, or low to high).	R/W <sup>1</sup>
3	—	This bit is read as 0. The write value should be 0.	R/W
4	PSRTO	PAUSE Frame Retransmit Over Flag 0: PAUSE frame retransmit count has not reached the upper limit 1: PAUSE frame retransmit count reached the upper limit.	R/W <sup>1</sup>

Bit	Symbol	Function	R/W
5	BFR	Continuous Broadcast Frame Reception Flag 0: Continuous reception of broadcast frames not detected 1: Continuous reception of broadcast frames detected.	R/W <sup>1</sup>
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Note 1. Write 1 to clear the flag.

The ECSR register indicates the status of the ETHERC. When any flag in the ECSR register is set to 1 while the associated bit in the ECSIPR register is 1 (interrupt enabled), the EDMAC0.EESR.ECI flag is set to 1.

### ICD flag (False Carrier Detect Flag)

The ICD flag indicates that the PHY-LSI has detected a false carrier on the line. The flag is set to 1 when a receive error signal shown in Figure 27.11 is received from the PHY-LSI. The information might not be correct when signals input from the PHY-LSI change faster than software recognizes the change. Check the timing of the PHY-LSI.

### LCHNG flag (Link Signal Change Flag)

The LCHNG flag indicates that the ET0\_LINKSTA signal input from the PHY-LSI has changed from high to low, or from low to high. Check the PSR.LMON flag for the current link status. See section 27.5.1. Preventing the LCHNG Flag from Erroneously Setting to 1 for more information.

### PSRTO flag (PAUSE Frame Retransmit Over Flag)

The PSRTO flag indicates that the number of retransmissions reached the value set in the TPAUSER register when retransmitting a PAUSE frame while automatic PAUSE frame transmission is enabled.

## 27.2.4 ECSIPR : ETHERC Interrupt Enable Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	BFSIP R	PSRT OIP	—	LCHN GIP	MPDIP	ICDIP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ICDIP	False Carrier Detect Interrupt Enable 0: Disable interrupt notification 1: Enable interrupt notification.	R/W
1	MPDIP	Magic Packet Detect Interrupt Enable 0: Disable interrupt notification 1: Enable interrupt notification.	R/W
2	LCHNGIP	LINK Signal Change Interrupt Enable 0: Disable interrupt notification 1: Enable interrupt notification.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	PSRTOIP	PAUSE Frame Retransmit Over Interrupt Enable 0: Disable interrupt notification 1: Enable interrupt notification.	R/W

Bit	Symbol	Function	R/W
5	BFSIPR	Continuous Broadcast Frame Reception Interrupt Enable 0: Disable interrupt notification 1: Enable interrupt notification.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The ECSIPR register selects whether to notify the EDMAC of the status indicated in the ECSR register. Each bit is associated with the flag with the same bit number in the ECSR register.

### 27.2.5 PIR : PHY Interface Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MDI	MDO	MMD	MDC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	x	0	0	0

Bit	Symbol	Function	R/W
0	MDC	MII/RMII Management Data Clock This value is output from the ET0_MDC pin to supply the management data clock to the MII or RMII.	R/W
1	MMD	MII/RMII Management Mode 0: Read 1: Write.	R/W
2	MDO	MII/RMII Management Data-Out This value is output from the ET0_MDIO pin when the MMD bit is 1 (write), and not when MMD is 0 (read).	R/W
3	MDI	MII/RMII Management Data-In This bit indicates the level of the ET0_MDIO pin. The write value should be 0.	R
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The PIR register accesses registers in the PHY-LSI through the MII or RMII. The management clock and management data are controlled by software. See [section 27.3.4. Accessing the MII and RMII Registers](#) for details on accessing the MII and RMII registers.

## 27.2.6 PSR : PHY Status Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LMON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	LMON	ET0_LINKSTA Pin Status Flag The link status can be read by connecting the link signal output from the PHY-LSI to the ET0_LINKSTA pin. For details on the polarity, see the specifications of the connected PHY-LSI.	R
31:1	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

The PSR register monitors interface signals from the PHY-LSI.

## 27.2.7 RDMLR : Random Number Generation Counter Upper Limit Setting Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0x40

Bit position:	31													19																		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RMD[19:0]																	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Note: S-TYPE3, P-TYPE3

Bit	Symbol	Function	R/W
19:0	RMD[19:0]	Random Number Generation Counter 0x00000: Normal operation 0x00001 to 0xFFFFF: Setting prohibited.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

The RDMLR register specifies the maximum value for the counter used in the random number generator. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled).

## 27.2.8 IPGR : Interpacket Gap Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0x50

Bit position:	31																											4				0		
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IPG[4:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

Bit	Symbol	Function	R/W
4:0	IPG[4:0]	Interpacket Gap 0x00: 16 bit times 0x01: 20 bit times ⋮ 0x14: 96 bit times (initial value) ⋮ 0x1F: 140 bit times.	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

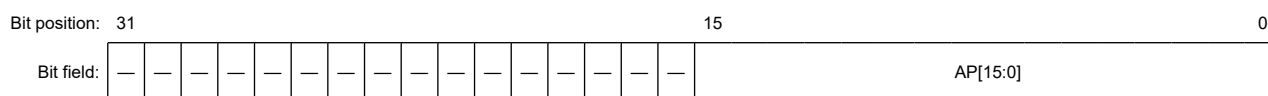
Note: S-TYPE3, P-TYPE3

The IPGR register specifies the interpacket gap (IPG) value. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled). See section 27.3.6. Adjusting Transmission Efficiency by Changing the IPG for details on the IPG.

### 27.2.9 APR : Automatic PAUSE Frame Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0x54



Value after reset: 0

Bit	Symbol	Function	R/W
15:0	AP[15:0]	Automatic PAUSE Time Setting These bits set the value of the pause_time parameter for PAUSE frames that are automatically transmitted. Transmission is not performed until a period equal to the set value multiplied by 512 bit times has elapsed.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

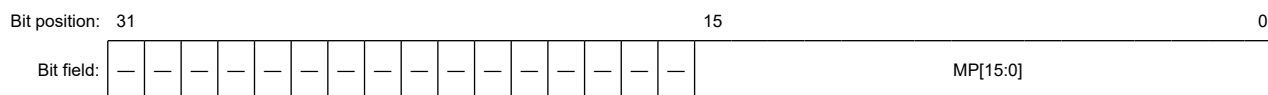
Note: S-TYPE3, P-TYPE3

The APR register specifies the PAUSE time for PAUSE frames that are automatically transmitted. The value set in the APR register is used for the pause\_time parameter of the PAUSE frame. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled).

### 27.2.10 MPR : Manual PAUSE Frame Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0x58



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 x x x x x x x x x x x x x x x x x

Bit	Symbol	Function	R/W
15:0	MP[15:0]	Manual PAUSE Time Setting These bits set the value of the pause_time parameter for PAUSE frames that are manually transmitted. Transmission is not performed until the set value multiplied by 512 bit times has elapsed. The read value is undefined.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3





Bit	Symbol	Function	R/W
31:0	MAHR[31:0]	MAC Address Upper Bit See the following.	R/W

Note: S-TYPE3, P-TYPE3

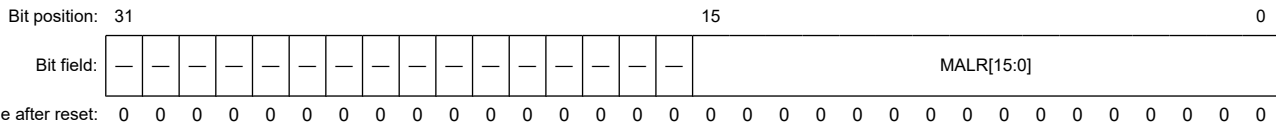
The MAHR register specifies the upper 32 bits ([47:16]) of the 48-bit MAC address. For example, if the MAC address is 01-23-45-67-89-AB, set the register to 0x01234567.

Set the MAHR register during initialization after a reset. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled). When rewriting this register, set the EDMAC0.EDMR.SWR bit to 1 to reset the EDMAC and ETHERC, then set this register again.

### 27.2.16 MALR : MAC Address Lower Bit Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0xC8



Bit	Symbol	Function	R/W
15:0	MALR[15:0]	MAC Address Lower Bit These bits set the lower 16 bits of the MAC address.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

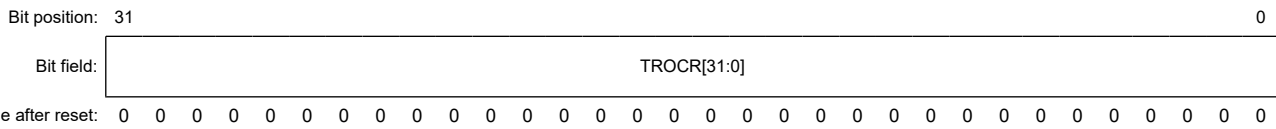
The MALR register specifies the lower 16 bits of the 48-bit MAC address. For example, if the MAC address is 01-23-45-67-89-AB, set the register to 0x000089AB.

Set the MALR register during initialization after a reset. Do not rewrite this register while the ECMR.TE bit is 1 (transmit function enabled) or while the ECMR.RE bit is 1 (receive function enabled). When rewriting this register, set the EDMAC0.EDMR.SWR bit to 1 to reset the EDMAC and ETHERC, then set this register again.

### 27.2.17 TROCR : Transmit Retry Over Counter Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0xD0



Bit	Symbol	Function	R/W
31:0	TROCR[31:0]	Transmit Retry Over Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The TROCR register is a counter that indicates the number of frames that failed to be retransmitted. The register value is incremented by 1 when a frame fails to be retransmitted 15 times. The counter stops when the register value becomes 0xFFFF\_FFFF. Writing any value to the TROCR register clears the counter value to 0.

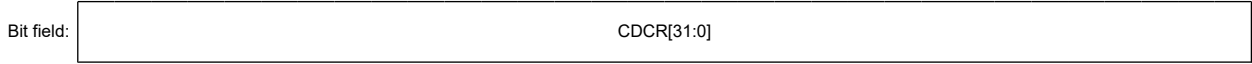


### 27.2.18 CDCR : Late Collision Detect Counter Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0xD4

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CDCR[31:0]	Late Collision Detect Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The CDCR register is a counter that indicates the number of late collisions that are detected after transmission starts. When the register value becomes 0xFFFF\_FFFF, the counter stops. Writing any value to the CDCR register clears the counter value to 0.

### 27.2.19 LCCR : Lost Carrier Counter Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0xD8

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	LCCR[31:0]	Lost Carrier Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

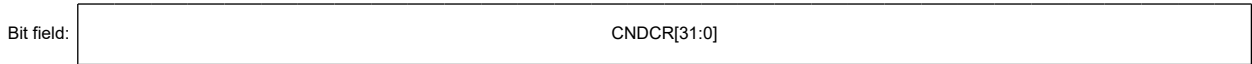
The LCCR register is a counter that indicates the number of times a loss of carrier is detected during frame transmission. When the register value becomes 0xFFFF FFFF, the counter stops. Writing any value to the LCCR register clears the counter value to 0.

### 27.2.20 CNDCCR : Carrier Not Detect Counter Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0xDC

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CNDCCR[31:0]	Carrier Not Detect Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The CNDCR register is a counter that indicates the number of times a carrier is not detected during preamble transmission. When the register value becomes 0xFFFF\_FFFF, the counter stops. Writing any value to the CNDCR register clears the counter value to 0.

### 27.2.21 CEFPCR : CRC Error Frame Receive Counter Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0xE4

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CEFCR[31:0]	CRC Error Frame Receive Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The CEFPCR register is a counter that indicates the number of received frames in which a CRC error was detected. When the register value becomes 0xFFFF\_FFFF, the counter stops. Writing any value to the CEFPCR register clears the counter value to 0.

### 27.2.22 FRECR : Frame Receive Error Counter Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0xE8

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	FRECR[31:0]	Frame Receive Error Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The FRECR register is a counter that indicates the number of times a frame receive error has occurred. The PHY-LSI notifies the ETHERC of the frame receive error using the ET0\_RX\_ER pin. The FRECR register increments each time the ET0\_RX\_ER pin goes high. When the register value becomes 0xFFFF\_FFFF, the counter stops. Writing any value to the FRECR register clears the counter value to 0.

### 27.2.23 TSFRCCR : Too-Short Frame Receive Counter Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0xEC

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	TSFRCCR[31:0]	Too-Short Frame Receive Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The TSFRCCR register is a counter that indicates the number of times a short frame that is shorter than 64 bytes was received. When the register value becomes 0xFFFF\_FFFF, the counter stops. Writing any value to the TSFRCCR register clears the counter value to 0.

### 27.2.24 TLFRCR : Too-Long Frame Receive Counter Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0xF0

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	TLFRCR[31:0]	Too-Long Frame Receive Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

Note: The TLFRCR register does not increment when a frame is received with an alignment error. In this case, the RFCR register increments.

The TLFRCR register is a counter that indicates the number of times a long frame that is longer than the RFLR register value was received. When the register value becomes 0xFFFF\_FFFF, the counter stops. Writing any value to the TLFRCR register clears the counter value to 0.

### 27.2.25 RFCR : Received Alignment Error Frame Counter Register

Base address: ETHERC0 = 0x4035\_4100  
ETHERC0\_NS = 0x5035\_4100

Offset address: 0xF4

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	RFCR[31:0]	Received Alignment Error Frame Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The RFCR register is a counter that indicates the number of times a frame was received with an alignment error, meaning that it is not an integral number of octets. When the register value becomes 0xFFFF\_FFFF, the counter stops. Writing any value to the RFCR register clears the counter value to 0.

### 27.2.26 MAFCR : Multicast Address Frame Receive Counter Register

Base address: ETHERC0 = 0x4035\_4100  
 ETHERC0\_NS = 0x5035\_4100

Offset address: 0xF8

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	MAFCR[31:0]	Multicast Address Frame Receive Counter See the following.	R/W

Note: S-TYPE3, P-TYPE3

The MAFCR register is a counter that indicates the number of times a frame with the multicast address set was received. When the register value becomes 0xFFFF\_FFFF, the counter stops. Writing any value to the MAFCR register clears the counter value to 0.

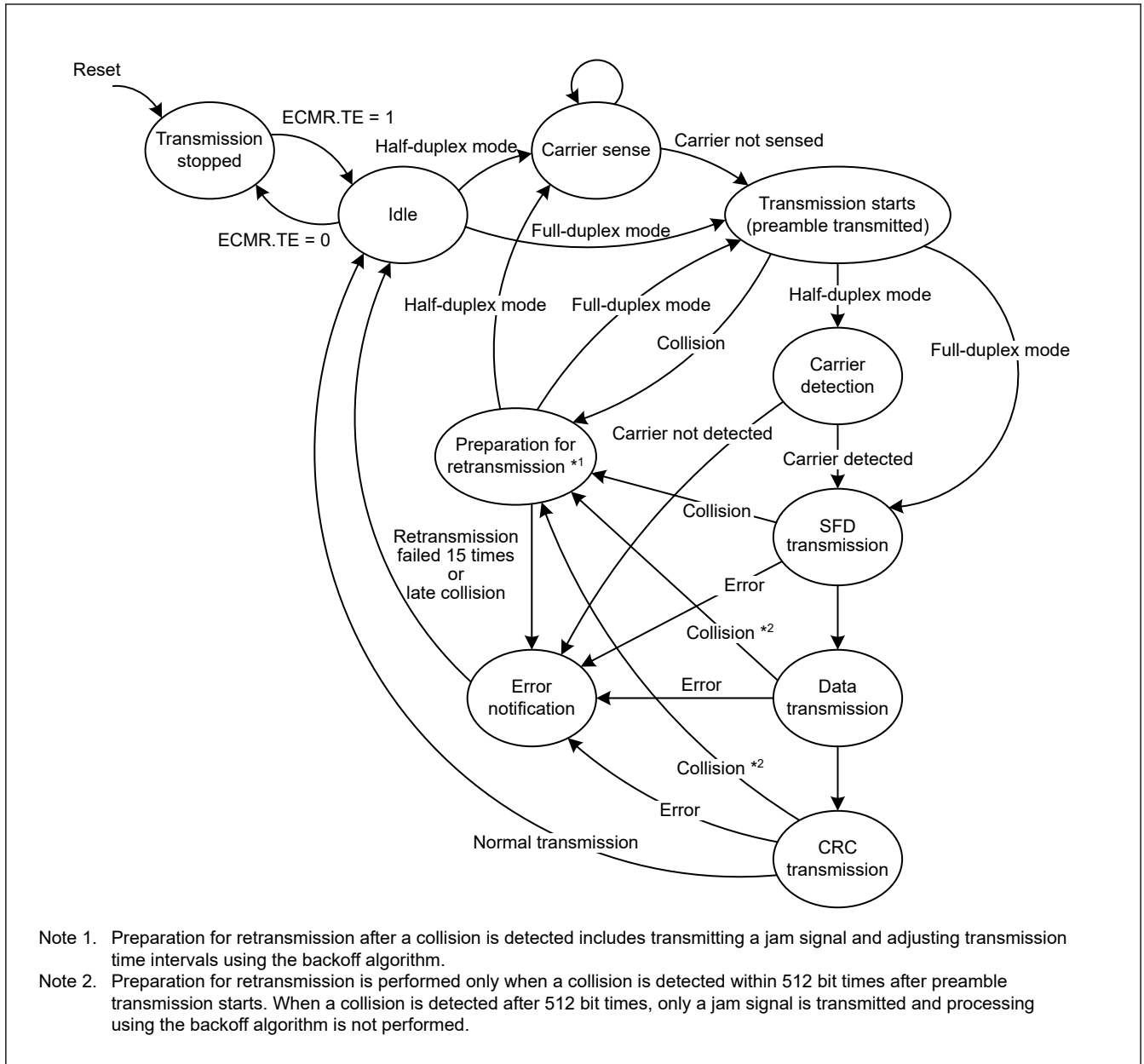
## 27.3 Operation

This section provides an overview of the ETHERC operations. ETHERC supports the flow control compliant with IEEE802.3x, and can transmit and receive PAUSE frames.

### 27.3.1 Transmission

The ETHERC transmitter assembles the transmit data into a frame and outputs it to the MII or RMII when a transmit request is received from the EDMAC. The frame transmitted through the MII or RMII is transmitted on the line by the PHY-LSI.

Figure 27.4 shows the state transitions of the ETHERC transmitter.



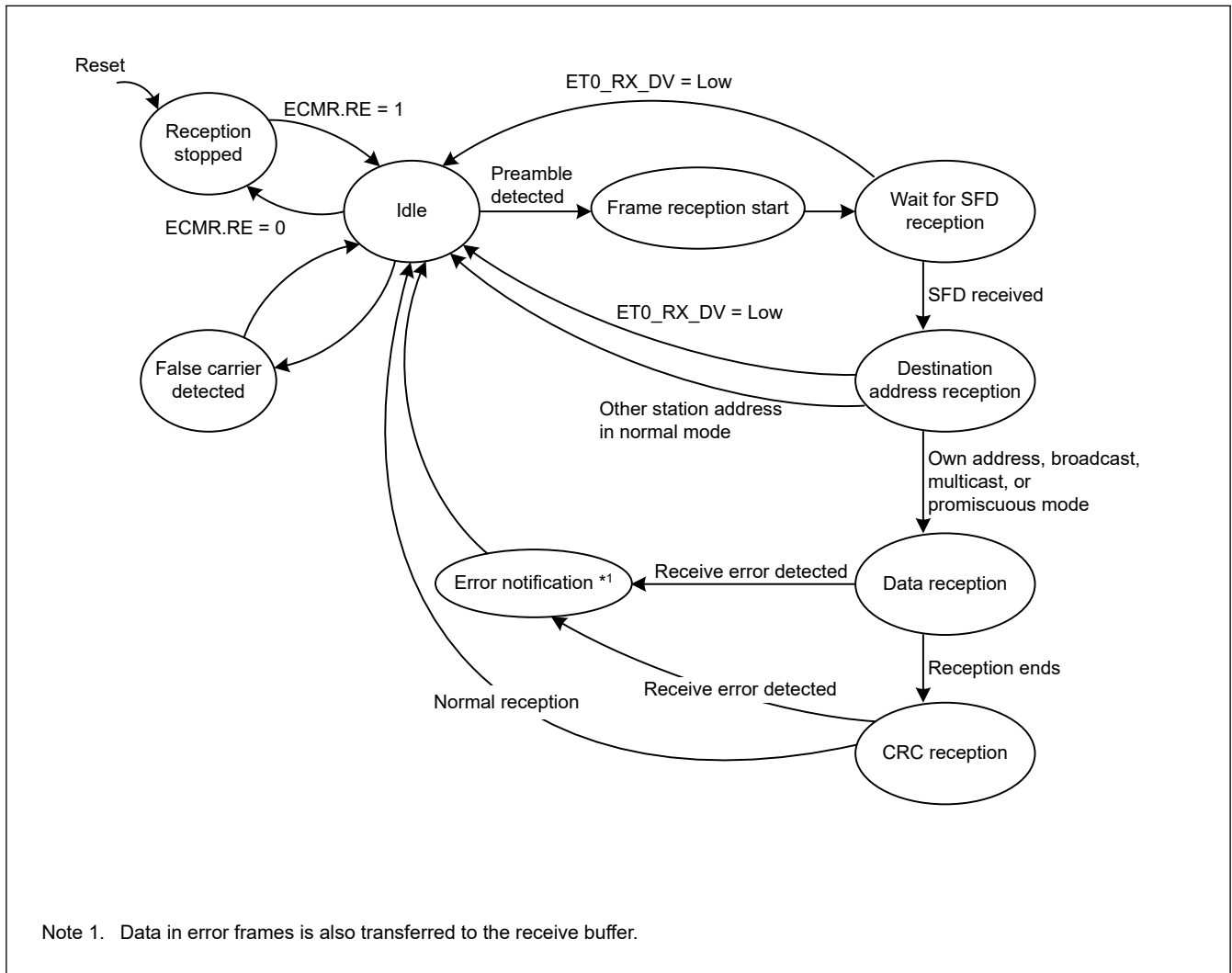
**Figure 27.4 ETHERC transmitter state transitions**

The ETHERC transmitter state transitions are as follows:

1. When the ECMR.TE bit is set to 1, the ETHERC enters the transmit idle state.
2. When a transmit request is received from the EDMAC, the ETHERC enters the carrier sense state. The ETHERC waits for the interpacket gap and then transmits a preamble to the MII or RMII. When full-duplex mode is selected, carrier sensing is not required, so the ETHERC transmits a preamble immediately after receiving a transmit request from the EDMAC.
3. The ETHERC transmits the Start Frame Delimiter (SFD), transmit data, and CRC sequentially. When the transmission completes successfully, the ETHERC notifies the EDMAC of successful completion, and the EDMAC sets the EDMAC0.EESR.TC flag to 1. When a late collision or loss of carrier is detected during data transmission, the ETHERC stops the transmission and notifies the EDMAC of the error.
4. After the time specified as the interpacket gap has elapsed, the ETHERC enters the idle state and continues the transmission when transmit data remains.

### 27.3.2 Reception

The ETHERC receiver separates the frame input from the MII or RMI into the preamble, SFD, receive data, and CRC, and transmits only the receive data (destination address, source address, type/length, data/LLC). Figure 27.5 shows the state transitions of the ETHERC receiver.



**Figure 27.5 ETHERC receiver state transitions**

The ETHERC receiver state transitions are as follows:

1. When the ECMR.RE bit is set to 1, the ETHERC enters the receive idle state.
2. When the SFD following the preamble of the receive packet is detected, the ETHERC starts reception. If the received SFD is invalid, the ETHERC discards the frame.
3. In normal mode, the ETHERC starts data reception when the destination address of the receive frame is the address of the MCU or the receive frame is a broadcast or multicast frame. In promiscuous mode, the ETHERC starts data reception regardless of the receive frame type.
4. After receiving data from the MII or RMI, the ETHERC performs a CRC check. The ETHERC notifies the EDMAC of the CRC check result. After the received data is transferred to the receive buffer, the CRC check result is written back to the receive descriptor as status. The result is also reflected in the EDMAC0.EESR.CERF flag.
5. When the ECMR.RE bit is 1 after one frame is received, the ETHERC prepares to receive the next frame.

### 27.3.3 Frame Timing

#### 27.3.3.1 MII frame timing

Figure 27.6 to Figure 27.11 show the MII frame timing.

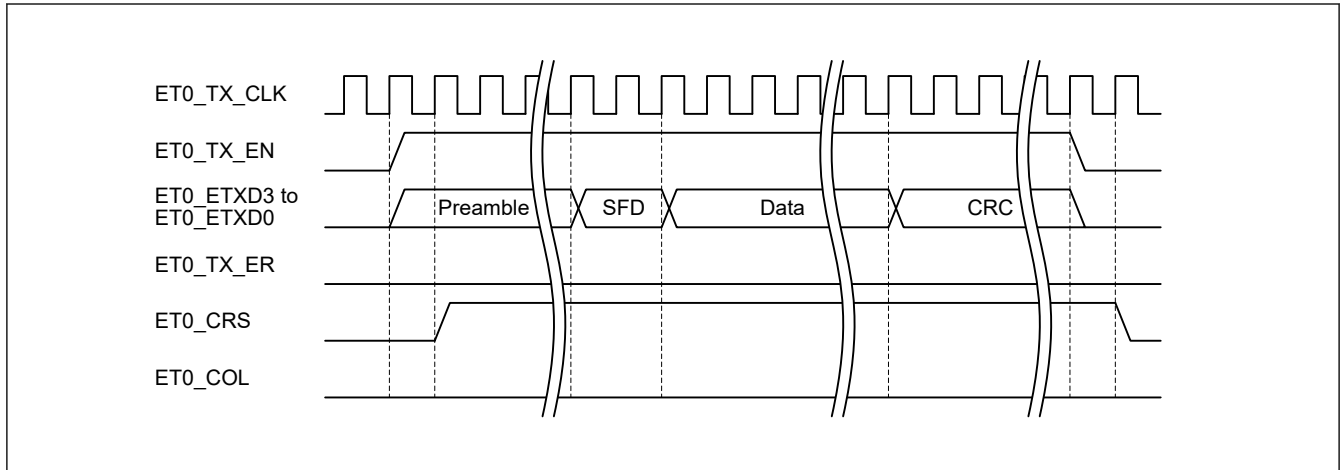


Figure 27.6 MII frame transmit timing during normal transmission

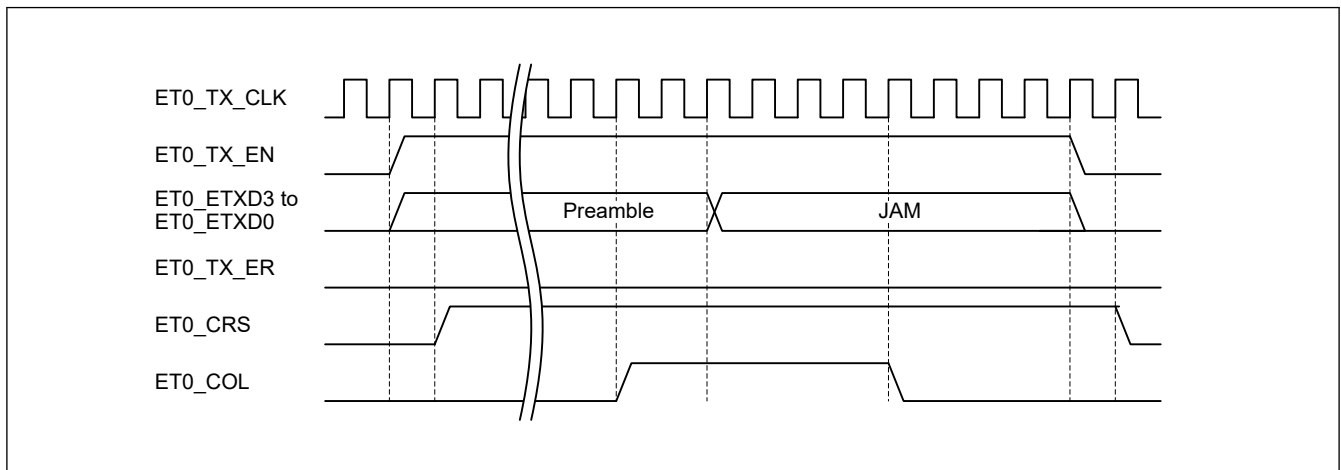


Figure 27.7 MII frame transmit timing when a collision occurs

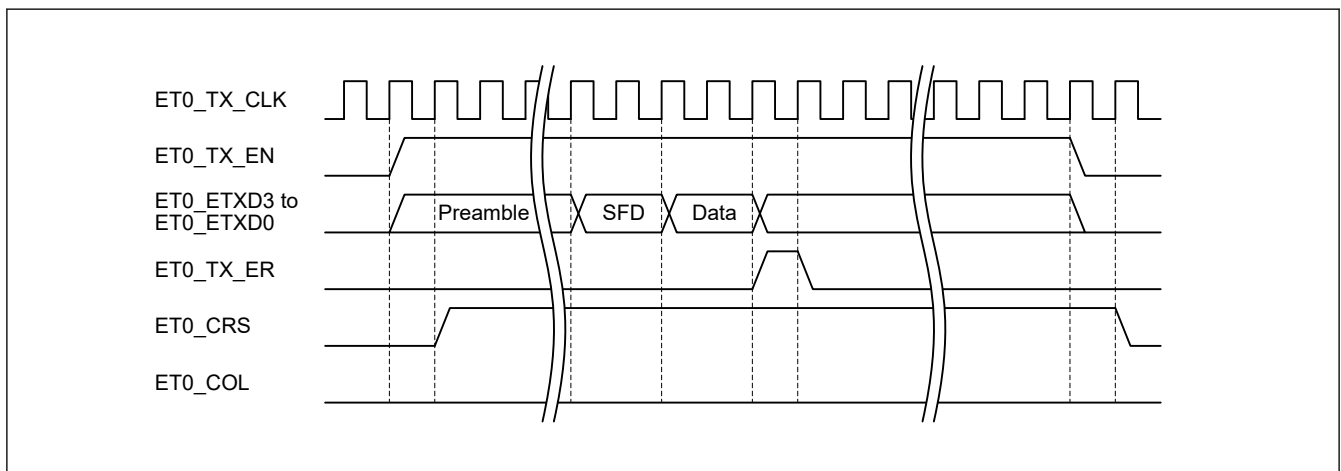


Figure 27.8 MII frame transmit timing when a transmit error occurs

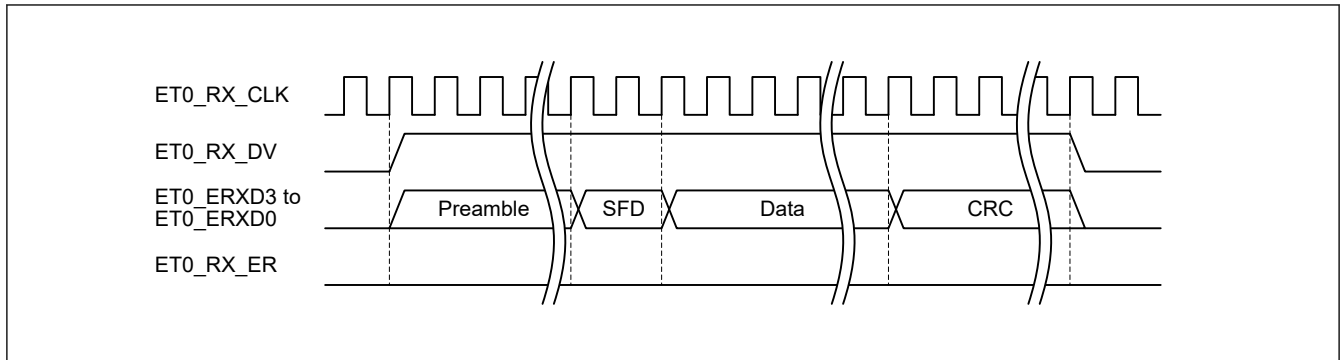


Figure 27.9 MII frame receive timing during normal reception

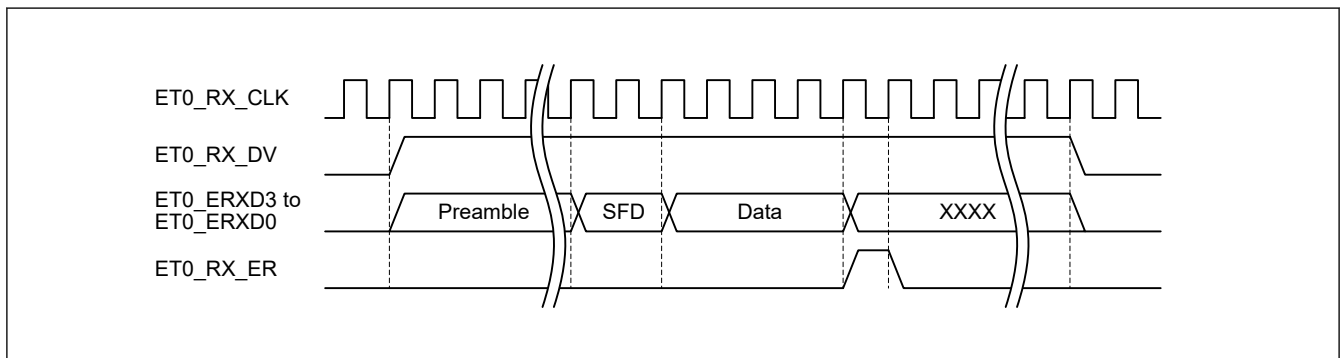


Figure 27.10 MII frame receive timing for receive error notification

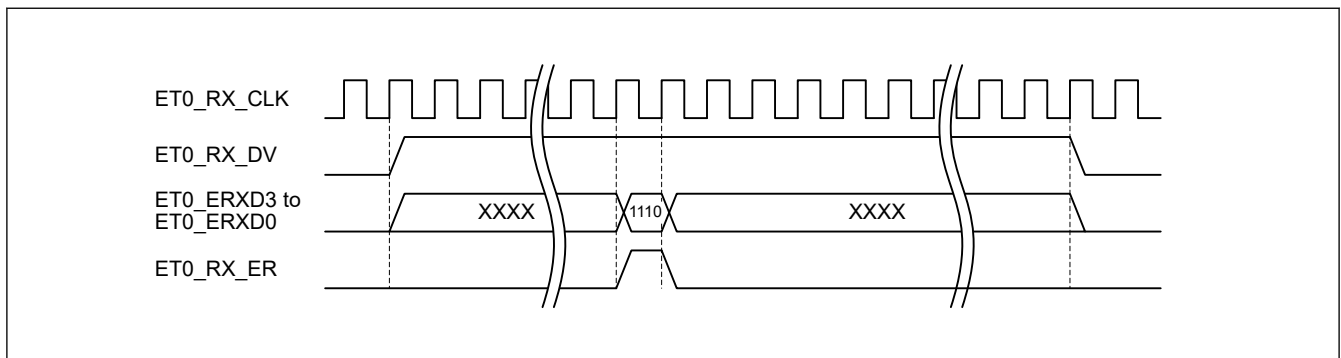


Figure 27.11 MII frame receive timing for false carrier notification

### 27.3.3.2 RMII frame timing

Figure 27.12 to Figure 27.14 show the RMII frame timing.

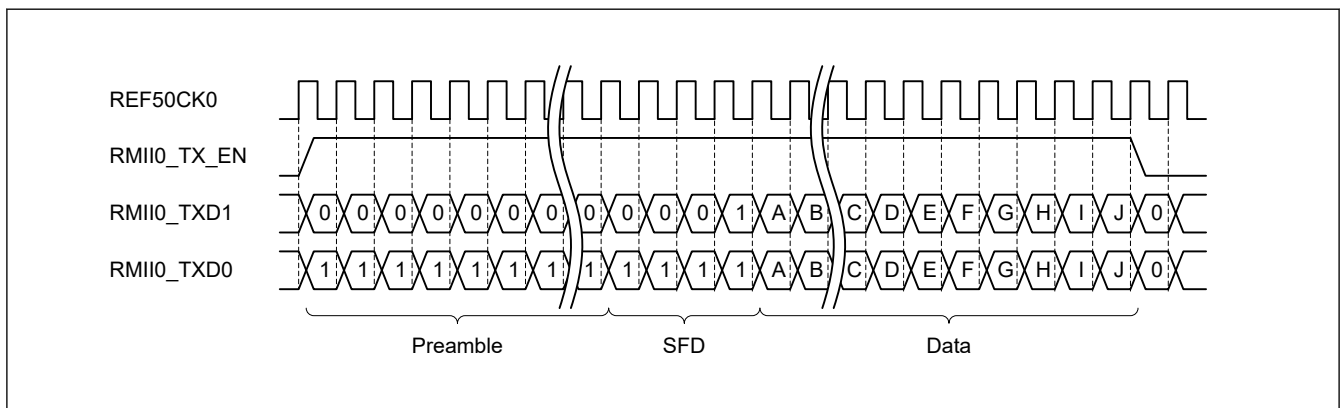


Figure 27.12 RMII frame transmit timing during normal transmission



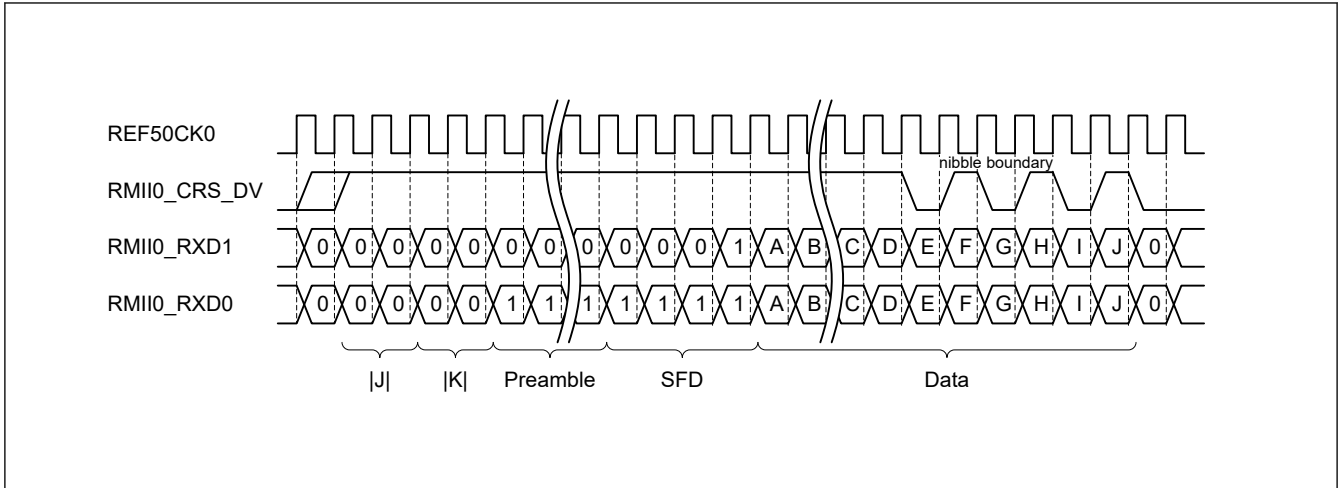


Figure 27.13 RMIi frame receive timing during normal reception

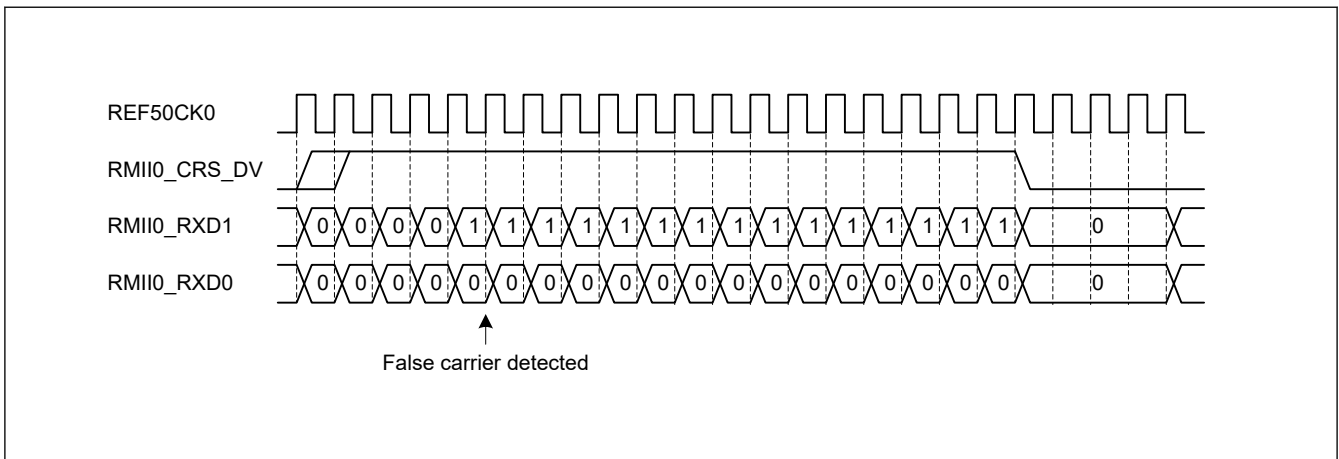


Figure 27.14 RMIi frame receive timing when a false carrier is detected

### 27.3.4 Accessing the MII and RMIi Registers

Use the PIR register to access the MII and RMIi registers in the PHY-LSI. Serial data in the MII and RMIi management frame format is transmitted and received through the ET0\_MDC and ET0\_MDIO pins controlled by software.

#### 27.3.4.1 MII and RMIi management frame format

Table 27.3 lists the MII and RMIi management frame formats.

Table 27.3 MII and RMIi management frame formats

Access type	MII and RMIi management frame								
	Parameter	PRE	ST	OP	PHYAD	REGAD	TA	DATA	IDLE
	Number of bits	32	2	2	5	5	2	16	1
Read		1...1	01	10	00001	RRRRR	Z0	DDDDDD DDDDDD DD	Z
Write		1...1	01	01	00001	RRRRR	10	DDDDDD DDDDDD DD	Z

Note: PRE (preamble): Send 32 consecutive 1s.  
 ST (start of frame): Send 01b.  
 OP (operation code): Send 10b for read or 01b for write.  
 PHYAD (PHY address): Up to 32 PHY-LSIs can be connected to one MAC. PHY-LSIs are selected with these 5 bits. When the PHY-LSI address is 1, send 0x01.

REGAD (register address): One register is selected from up to 32 registers in the PHY-LSI. When the register address is 1, send 0x01.  
 TA (turnaround): Use 2-bit turnaround time to avoid contention between the register address and data during a read operation. Send 10b during a write operation. Release the bus for 1 bit during a read operation (Z is output). (This is indicated as Z0 because 0 is output from the PHY-LSI on the next clock cycle.)  
 DATA (data): 16-bit data. Sequentially send or receive starting from the MSB.  
 IDLE (IDLE condition): Wait time before inputting the next MII or RMII management format. Release the bus during a write operation (Z is output). No control is required, because a bus was already released during a read operation.

### 27.3.4.2 MII and RMII register access procedure

Access to the MII and RMII registers includes writing data in 1-bit units, reading data in 1-bit units, and releasing the bus. Figure 27.15 to Figure 27.18 show examples of the MII and RMII register access timing. The access timing differs with the PHY-LSI type.

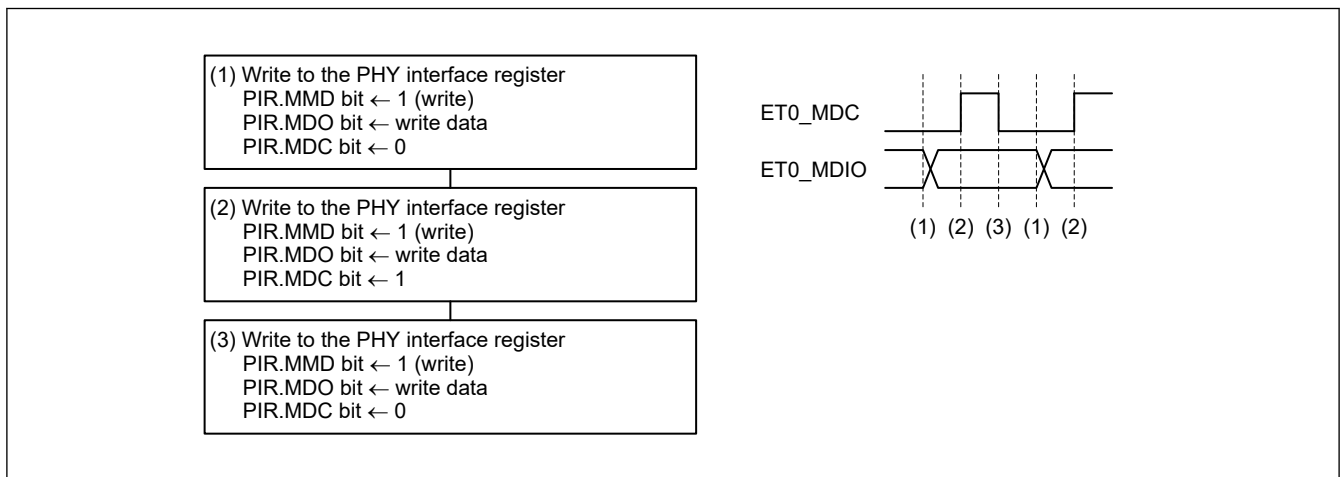


Figure 27.15 1-bit data write flow

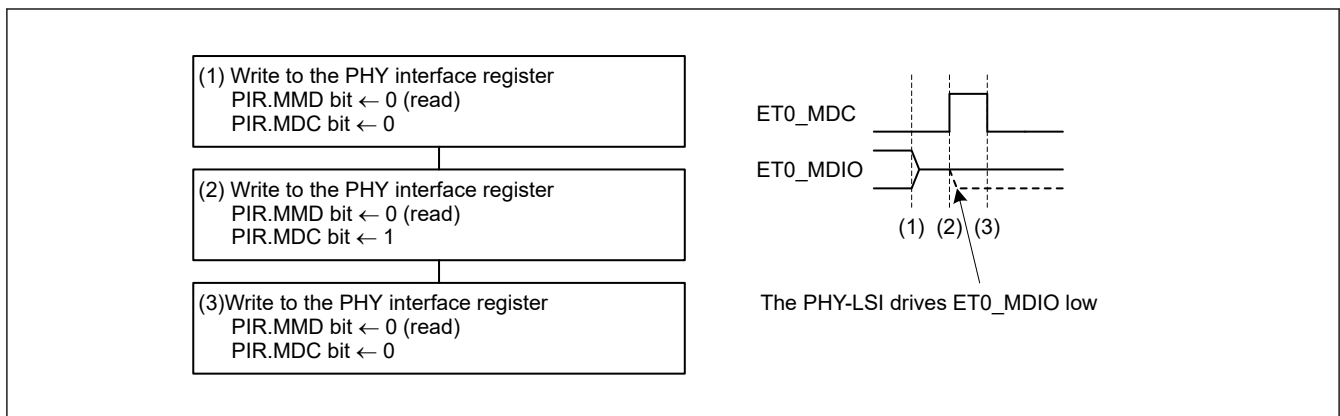


Figure 27.16 Bus release flow, with TA in read operation in Table 27.3

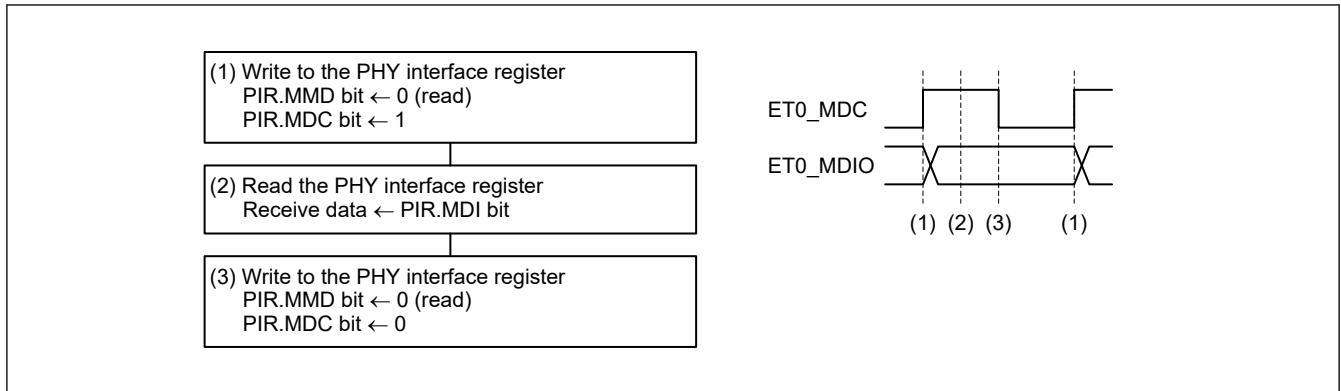


Figure 27.17 1-bit data read flow

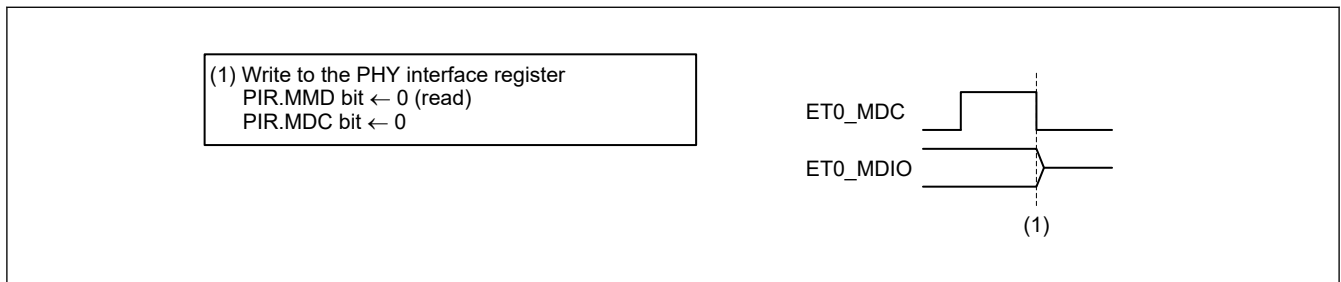


Figure 27.18 Bus release flow, with IDLE in write operation in Table 27.3

### 27.3.5 Magic Packet Detection

The ETHERC supports Wake-on-LAN (WOL). WOL is a function to detect a Magic Packet transmitted from a host device or other device, and to wake the MCU from a low power mode such as Sleep. When the ETHERC detects a Magic Packet, it outputs high on the ET0\_WOL pin. Write 1 to the EDMAC0.EDMR.SWR bit to drive the ET0\_WOL pin low.

Because a Magic Packet is transmitted in broadcast mode, it is received regardless of the destination MAC address selected in the format. The ETHERC outputs high on the ET0\_WOL pin only when the destination MAC address matches its own MAC address. See the technical documentation provided by Advanced Micro Devices, Inc., for details on the Magic Packet.

To use WOL in the MCU, use the procedure in the following example:

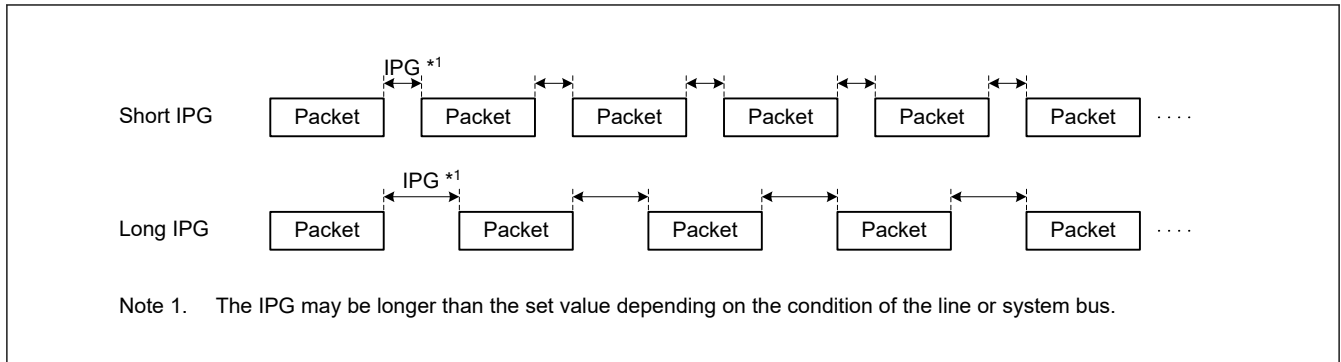
1. Configure the ICU to disable ETHER\_EINT0 interrupt requests.
2. Set the ECMR.MPDE bit to 1 to enable Magic Packet detection, and set the ECMR.RE bit to 1 to enable reception.
3. Set the ECSIPR.MPDIP bit to 1 to enable notification of Magic Packet detection interrupts.
4. Set the EDMAC0.EESIPR.ECIIP bit to 1 to enable ETHERC status register source interrupts.
5. Configure the ICU to enable ETHER\_EINT0 interrupt requests.
6. Change the CPU operating mode to CPU Sleep mode or place unused peripherals in the module-stop state, as required.
7. When a Magic Packet is detected, an interrupt request is sent to the CPU. High is output on the ET0\_WOL pin to notify peripheral devices that the Magic Packet was detected.

#### 27.3.5.1 Constraints on Magic Packet detection

The ETHERC receives packets, including broadcast packets, even when waiting to receive a Magic Packet. This means that receive data might already be stored in the receive FIFO of the EDMAC when a Magic Packet is detected. Also, flags in the ECSR and EDMAC0.EESR registers might have changed. When returning to normal operation after detecting a Magic Packet, set the EDMAC0.EDMR.SWR bit to 1 to reset the ETHERC and EDMAC.

### 27.3.6 Adjusting Transmission Efficiency by Changing the IPG

The IPG is a non-transmit period between transmit frames. The ETHERC can change the value of the IPG to increase or decrease transmission efficiency based on the value set in the IPGR register. Typical values are specified in the IEEE802.3 standard. When changing the setting, confirm that all devices in the same network operate normally.



**Figure 27.19 Differences in transmission efficiency based on changes in the IPG**

### 27.3.7 Flow Control

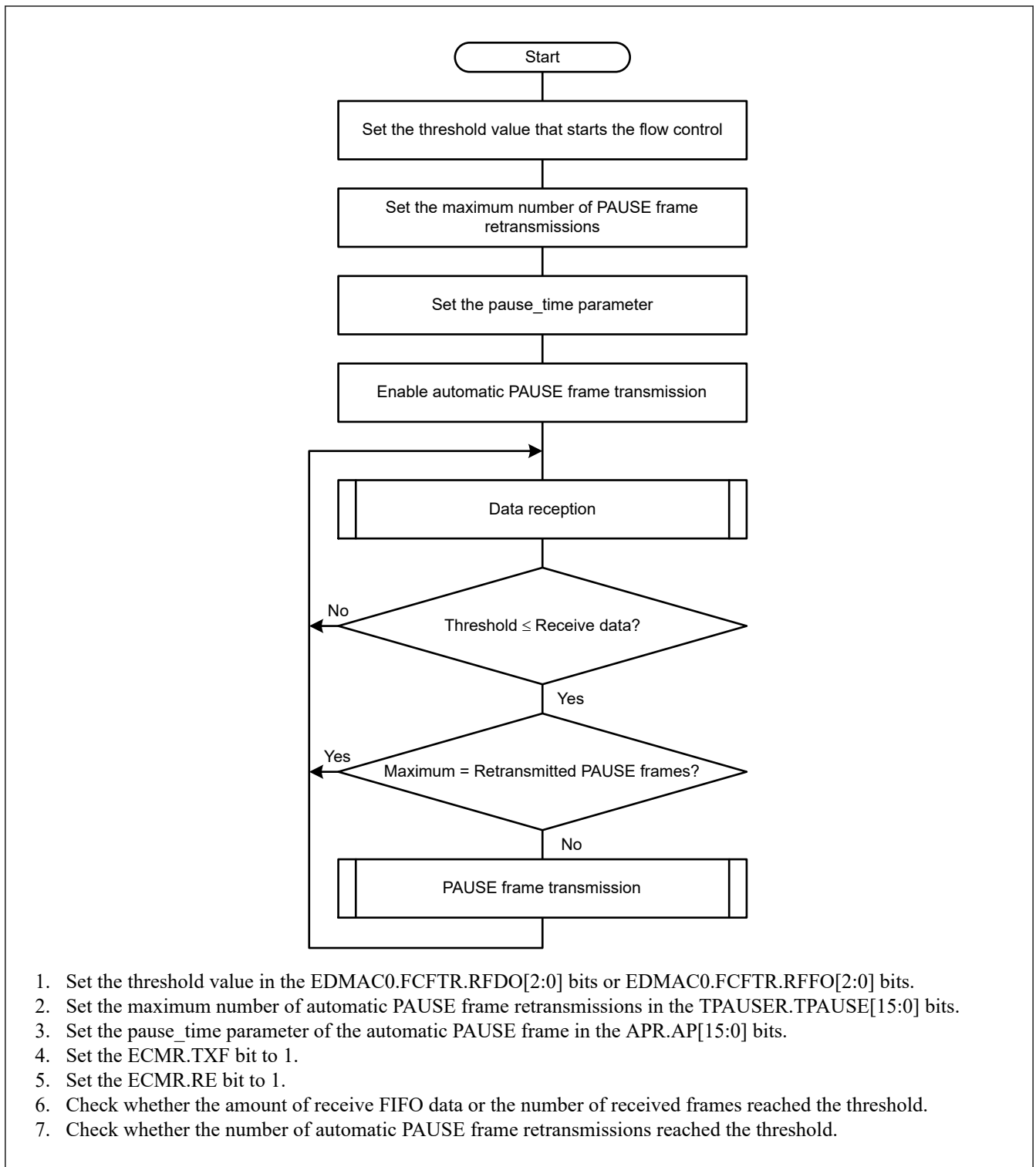
The ETHERC can perform flow control compliant with IEEE802.3x in full-duplex mode, and the receiver and transmitter can be set independently. PAUSE frames can be transmitted automatically or manually.

#### 27.3.7.1 Automatic PAUSE frame transmission

When the ECMR.TXF bit is set to 1, automatic PAUSE frame transmission is enabled. A PAUSE frame is automatically transmitted by a PAUSE frame transmit request from the EDMAC. The APR.AP[15:0] bit value is used for the `pause_time` parameter of the PAUSE frame.

When a PAUSE frame is transmitted, if the EDMAC is still requesting PAUSE frame transmission after the PAUSE time elapses, a PAUSE frame is transmitted again. The maximum number of PAUSE frame retransmissions can be set in the TPAUSER.TPAUSE[15:0] bits. If the maximum number of retransmissions is reached, subsequent PAUSE frames are not transmitted.

[Figure 27.20](#) shows the procedure for setting up automatic PAUSE frame transmission.



**Figure 27.20 Example procedure for setting up automatic PAUSE frame transmission**

### 27.3.7.2 Manual PAUSE frame transmission

A PAUSE frame can be manually transmitted at any time. When the software writes the pause\_time parameter of the PAUSE frame to the MPR.MP[15:0] bits, the ETHERC transmits a PAUSE frame once. To transmit a PAUSE frame more than once, write to the MPR.MP[15:0] bits for each transmission.

### 27.3.7.3 PAUSE frame reception

When the ECMR.RXF bit is set to 1, PAUSE frame detection is enabled. After a PAUSE frame is received, the ETHERC completes transmission of the current frame and waits for the PAUSE time of the received PAUSE frame to elapse before it can transmit the next frame. The ETHERC also increments the RFCF.RPAUSE[7:0] bit value.

However, while waiting for the PAUSE time to elapse, if a PAUSE frame that contains a pause\_time parameter of 0 is received and the ECMR.ZPF bit is 1, the ETHERC becomes ready to transmit immediately.

## 27.4 Interrupts

When a flag in the ECSR register sets to 1 and the associated bit in the ECSIPR register is 1, the ETHERC notifies the EDMAC of the interrupt source status. After receiving the notification, the EDMAC sets the EDMAC0.EESR.ECI flag to 1. When the EDMAC0.EESIPR.ECIIP bit is 1, the EDMAC sends an ETHER\_EINT0 interrupt request to the CPU.

For details, see [section 28, Ethernet DMA Controller \(EDMAC\)](#).

## 27.5 Usage Notes

### 27.5.1 Preventing the LCHNG Flag from Erroneously Setting to 1

The ECSR.LCHNG flag might set to 1 even when the input level of the ET0\_LINKSTA pin remains the same. In this case, high level is input to the ET0\_LINKSTA pin when setting the PFS.PmnPFS register to assign the ET0\_LINKSTA signal to a port, or when releasing the ETHERC and EDMAC software reset using the EDMAC0.EDMR.SWR bit. The ECSR.LCHNG flag is set to 1 because the ET0\_LINKSTA signal in the ETHERC is fixed low level regardless of the input level to the external pin if the MPC does not assign the ET0\_LINKSTA signal or during an ETHERC and EDMAC software reset.

To avoid erroneously generating a link signal change interrupt, clear the ECSR.LCHNG flag, and then set the ECSIPR.LCHNGIP bit to 1.

### 27.5.2 Input to RMII0\_RX\_ER Pin While RMII Is Selected

When the width of a reception error signal received from the PHY-LSI is only 1 cycle of the REF50CK0 clock (50 MHz) while the RMII is selected, the signal is not recognized as an error signal.

### 27.5.3 Collision Occurrence in Half-Duplex Mode

Transmission might start and communication might collide within 21 clock cycles (50 MHz) from reception in halfduplex mode.

## 28. Ethernet DMA Controller (EDMAC)

### 28.1 Overview

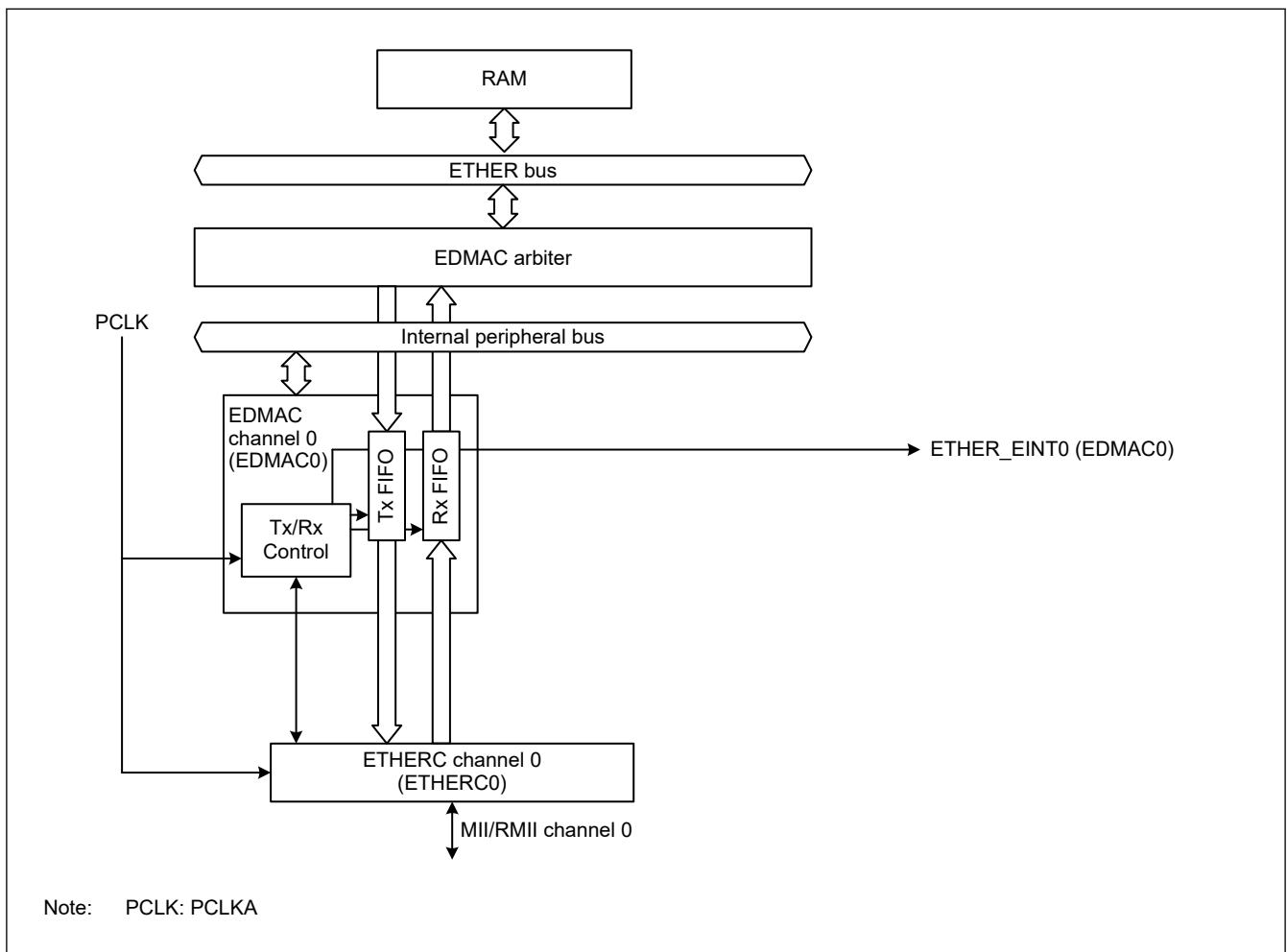
The MCU provides one channel for the Ethernet DMA Controller (EDMAC) for the Ethernet Controller (ETHERC).

The EDMAC controls most of the transmit and receive buffer management for communications. This reduces the load on the CPU and allows efficient data transmission and reception. The data transfers are controlled according to the information referred to as descriptors, in memory. EDMAC0 controls data transmission and reception for ETHERC0.

Table 28.1 lists the EDMAC specifications and Figure 28.1 shows the configuration. Figure 28.2 shows the configuration of descriptors and the transmit and receive buffers in memory.

**Table 28.1 EDMAC specifications**

Parameter	Specifications
Data transmission and reception	<ul style="list-style-type: none"> <li>Controls data transmission and reception according to descriptors</li> <li>Supports single buffer frame transmission and reception (1 buffer per frame) and multi-buffer frame transmission and reception (multiple buffers per frame)</li> </ul>
Functions	<ul style="list-style-type: none"> <li>Minimizes system bus occupancy time using block transfer (32-byte units)</li> <li>Writes back the transmit or receive frame state to descriptors</li> <li>Inserts padding in receive data</li> </ul>
Module-stop function	Module-stop state can be set to reduce power consumption
Trust Zone Filter	Security and Privilege attribution can be set



**Figure 28.1 EDMAC configuration**

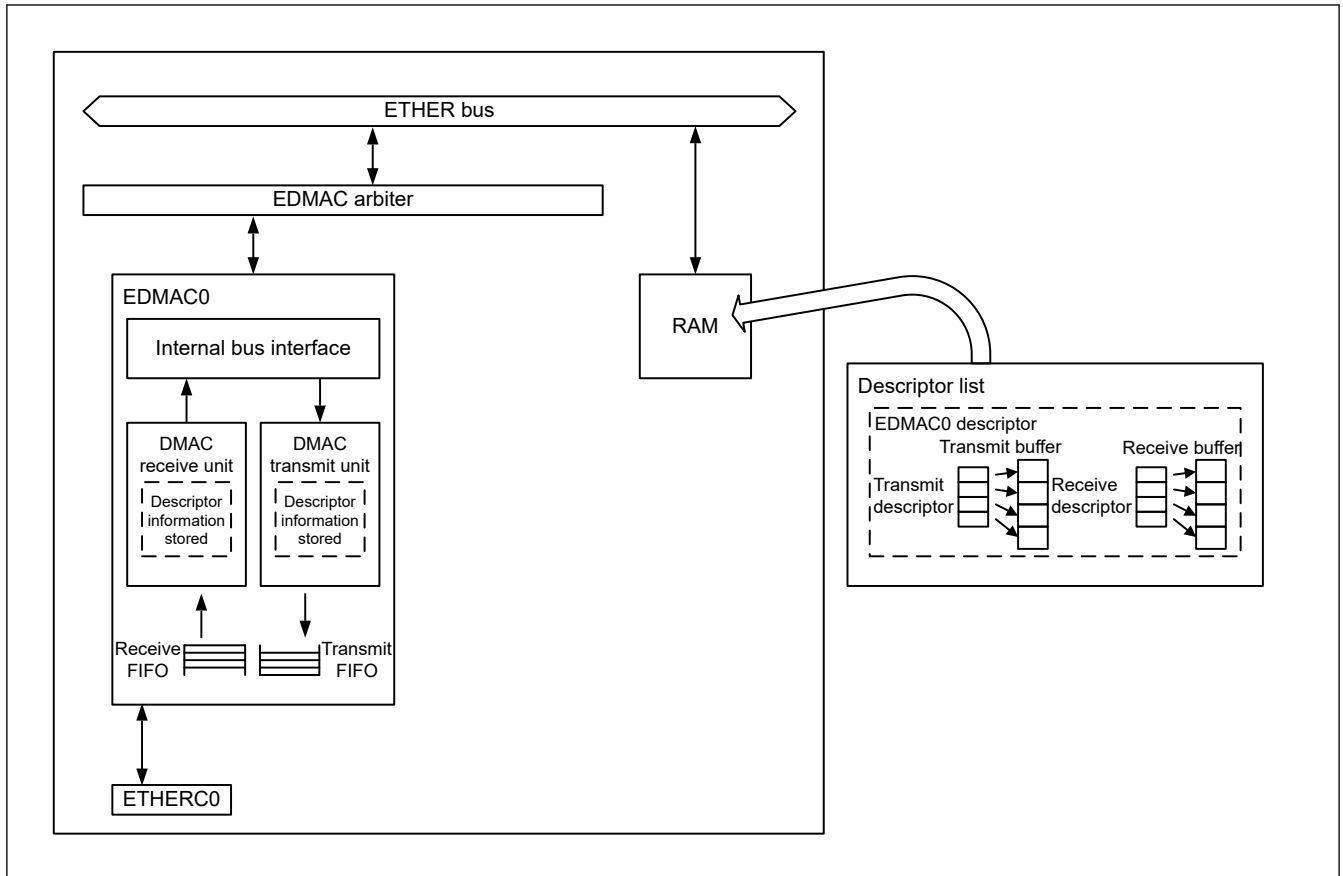


Figure 28.2 Configuration of descriptors and transmit and receive buffers in memory

## 28.2 Register Descriptions

### 28.2.1 EDMR : EDMAC Mode Register

Base address: EDMAC0 = 0x4035\_4000  
 EDMAC0\_NS = 0x5035\_4000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	DE	DL[1:0]	—	—	—	—	SWR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SWR	Software Reset When 1 is written, the associated channels of the EDMAC and ETHERC are reset. The TDLAR, RDLAR, RMFCR, TFUCR, and RFOCR registers are not reset with this bit. The read value is 0.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W



Bit	Symbol	Function	R/W
5:4	DL[1:0]	Transmit/Receive Descriptor Length 0 0: 16 bytes 0 1: 32 bytes 1 0: 64 bytes 1 1: 16 bytes.	R/W
6	DE	Big Endian Mode/Little Endian Mode* <sup>1</sup> 0: Big endian mode 1: Little endian mode.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Note 1. This setting applies to data for the transmit and receive buffers. It does not apply to transmit and receive descriptors and registers.

The EDMR register controls EDMAC operation. Set the EDMR register during initialization process after a reset. When rewriting this register outside of the initialization process, set the SWR bit to 1 to reset the EDMAC and ETHERC, then set this register again. If the ETHERC and EDMAC are reset during data transmission or reception, abnormal data might be sent on the line. Do not rewrite this register while the ETHERC transmit or receive function is enabled. It takes 64 cycles of the peripheral module clock (PCLKA) to initialize the ETHERC and EDMAC. Complete the initialization before accessing registers in the ETHERC and EDMAC.

### 28.2.2 EDTRR : EDMAC Transmit Request Register

Base address: EDMAC0 = 0x4035\_4000  
EDMAC0\_NS = 0x5035\_4000

Offset address: 0x08

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

Bit	Symbol	Function	R/W
0	TR	Transmit Request When 1 is written, the EDMAC reads the associated descriptor and transmits frames where the TD0.TACT bit is 1. The TR bit clears to 0 after all the valid frames are transmitted. Writing 0 to this bit has no effect.	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The EDTRR register controls EDMAC transmission. After the EDMAC transmits one frame, it reads the next descriptor. When the TD0.TACT bit in the descriptor is 1, the EDMAC continues transmission. When the TD0.TACT bit is 0, the EDMAC sets the TR bit to 0 and stops transmission.

### 28.2.3 EDRRR : EDMAC Receive Request Register

Base address: EDMAC0 = 0x4035\_4000  
EDMAC0\_NS = 0x5035\_4000

Offset address: 0x10

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

Bit	Symbol	Function	R/W
0	RR	Receive Request 0: Disable the receive function*1 1: Read receive descriptor and enable the receive function.	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Note 1. If the receive function is disabled during frame reception, write-back to the receive descriptor is not performed successfully. Subsequent pointers for reading a receive descriptor become abnormal and the EDMAC cannot operate normally. In this case, to enable the EDMAC receive function again, execute a software reset by setting the EDMR.SWR bit to 1. To disable the EDMAC receive function without resetting the EDMAC, set the ETHERC0.ECMR.RE bit to 0. After the EDMAC completes reception and write-back to the receive descriptor is confirmed, set the RR bit to 0.

The EDRRR register controls EDMAC reception. When the RR bit sets to 1, the EDMAC reads the receive descriptor.

When the RD0.RACT bit is 1, the EDMAC waits for a receive request from the ETHERC. When the EDMAC has received data for the receive buffer size, it reads the next descriptor and waits to receive a frame. If the RD0.RACT bit is 0, the EDMAC sets the RR bit to 0 and stops reception.

### 28.2.4 TDLAR : Transmit Descriptor List Start Address Register

Base address: EDMAC0 = 0x4035\_4000  
EDMAC0\_NS = 0x5035\_4000

Offset address: 0x18

Bit position:	31																0
Bit field:	[Empty field]																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	—	These bits specify the start address of the transmit descriptor list. Set the start address according to the descriptor length selected in the EDMR.DL[1:0] bits. <ul style="list-style-type: none"> <li>• 16-byte boundary: Lower 4 bits = 0000b</li> <li>• 32-byte boundary: Lower 5 bits = 00000b</li> <li>• 64-byte boundary: Lower 6 bits = 000000b.</li> </ul>	R/W

Note: S-TYPE3, P-TYPE3

The TDLAR register specifies the start address of the transmit descriptor list. Align each descriptor on the associated boundary to the descriptor length selected in the EDMR.DL[1:0] bits. Do not rewrite the TDLAR register during transmission. Rewrite the TDLAR register while the EDTRR.TR bit is 0.

### 28.2.5 RDLAR : Receive Descriptor List Start Address Register

Base address: EDMAC0 = 0x4035\_4000  
 EDMAC0\_NS = 0x5035\_4000

Offset address: 0x20

Bit position: 31 0

Bit field:

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	—	The start address of the receive descriptor list is set. Set the start address according to the descriptor length selected in the EDMR.DL[1:0] bits. <ul style="list-style-type: none"> <li>• 16-byte boundary: Lower 4 bits = 0000b</li> <li>• 32-byte boundary: Lower 5 bits = 00000b</li> <li>• 64-byte boundary: Lower 6 bits = 000000b.</li> </ul>	R/W

Note: S-TYPE3, P-TYPE3

The RDLAR register specifies the start address of the receive descriptor list. Allocate each descriptor on the associated boundary to the descriptor length selected in the EDMR.DL[1:0] bits. Do not rewrite the RDLAR register during reception. Rewrite the RDLAR register while the EDRRR.RR bit is 0.

### 28.2.6 EESR : ETHERC/EDMAC Status Register

Base address: EDMAC0 = 0x4035\_4000  
 EDMAC0\_NS = 0x5035\_4000

Offset address: 0x28

Bit position: 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

Bit field:

—	TWB	—	—	—	TABT	RABT	RFCO <sub>F</sub>	ADE	ECI	TC	TDE	TFUF	FR	RDE	RFOF
---	-----	---	---	---	------	------	-------------------	-----	-----	----	-----	------	----	-----	------

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

—	—	—	—	CND	DLC	CD	TRO	RMAF	—	—	RRF	RTL <sub>F</sub>	RTS <sub>F</sub>	PRE	CER <sub>F</sub>
---	---	---	---	-----	-----	----	-----	------	---	---	-----	------------------	------------------	-----	------------------

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	CER <sub>F</sub>	CRC Error Flag 0: CRC error not detected 1: CRC error detected.	R/W
1	PRE	PHY-LSI Receive Error Flag 0: PHY-LSI receive error not detected 1: PHY-LSI receive error detected.	R/W
2	RTS <sub>F</sub>	Frame-Too-Short Error Flag 0: Frame-too-short error not detected 1: Frame-too-short error detected.	R/W
3	RTL <sub>F</sub>	Frame-Too-Long Error Flag 0: Frame-too-long error not detected 1: Frame-too-long error detected.	R/W
4	RRF	Alignment Error Flag 0: Alignment error not detected 1: Alignment error detected.	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	RMAF	Multicast Address Frame Receive Flag 0: Multicast address frame not received 1: Multicast address frame received.	R/W
8	TRO	Transmit Retry Over Flag 0: Transmit retry-over condition not detected 1: Transmit retry-over condition detected.	R/W
9	CD	Late Collision Detect Flag 0: Late collision not detected 1: Late collision detected during frame transmission.	R/W
10	DLC	Loss of Carrier Detect Flag 0: Loss of carrier not detected 1: Loss of carrier detected during frame transmission.	R/W
11	CND	Carrier Not Detect Flag 0: Carrier detected when transmission started 1: Carrier not detected during preamble transmission.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	RFOF	Receive FIFO Overflow Flag 0: No overflow occurred 1: Overflow occurred.	R/W
17	RDE	Receive Descriptor Empty Flag 0: EDMAC detected that the receive descriptor valid bit (RD0.RACT) is 1 1: EDMAC detected that the receive descriptor valid bit (RD0.RACT) is 0.	R/W
18	FR	Frame Receive Flag 0: Frame not received 1: Frame received and update of the receive descriptor is complete.	R/W
19	TFUF	Transmit FIFO Underflow Flag 0: No underflow occurred 1: Underflow occurred.	R/W
20	TDE	Transmit Descriptor Empty Flag 0: EDMAC detected that the transmit descriptor valid bit (TD0.TACT) is 1 1: EDMAC detected that the transmit descriptor valid bit (TD0.TACT) is 0.	R/W
21	TC	Frame Transfer Complete Flag 0: Transfer not complete or no transfer requested 1: All frames indicated in the transmit descriptor were completely transferred to the transmit FIFO.	R/W
22	ECI	ETHERC Status Register Source Flag 0: ETHERC status interrupt source not detected 1: ETHERC status interrupt source detected.	R <sup>*1</sup>
23	ADE	Address Error Flag 0: Invalid memory address not detected (normal operation) 1: Invalid memory address detected.* <sup>2</sup>	R/W
24	RFCOF	Receive Frame Counter Overflow Flag 0: Receive frame counter did not overflow 1: Receive frame counter overflowed.	R/W
25	RABT	Receive Abort Detect Flag 0: Frame reception not aborted or no reception requested 1: Frame reception aborted.	R/W
26	TABT	Transmit Abort Detect Flag 0: Frame transmission not aborted or no transmission requested. 1: Frame transmission aborted.	R/W
29:27	—	These bits are read as 0. The write value should be 0.	R/W
30	TWB	Write-Back Complete Flag 0: Write-back not complete or no transmission requested 1: Write-back to the transmit descriptor completed.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Note 1. The ECI flag is read-only. When the source in the ECSR register is cleared, the ECI flag is also cleared.

Note 2. When an address error is detected, the EDMAC halts the process. To resume operation, set the EDMR.SWR bit to 1 (resetting the EDMAC and ETHERC), and then reconfigure the EDMAC and ETHERC.

The EESR register indicates the ETHERC and EDMAC communication status. Each flag in the EESR register can be output as an interrupt request signal (ETHER\_EINT0) from the EDMAC. Writing 1 clears all of the flags except ECI to 0. Writing 0 does not affect any of the flag values. The interrupt sources are enabled by setting the associated bits in the EESIPR register.

#### **CERF flag (CRC Error Flag)**

The CERF flag sets to 1 when an error is detected while checking the frame check sequence (FCS) field of the receive frame.

#### **PRE flag (PHY-LSI Receive Error Flag)**

The PRE flag indicates that the RX\_ER signal output from the PHY-LSI is high.

#### **RTSF flag (Frame-Too-Short Error Flag)**

The RTSF flag indicates that a received frame is less than 64 bytes.

#### **RTLFL flag (Frame-Too-Long Error Flag)**

The RTLFL flag indicates that a received frame is greater than the upper limit of the receive frame length set in the ETHERC0.RFLR register. The excess data is discarded.

#### **RRF flag (Alignment Error Flag)**

The RRF flag indicates that a frame is not an integral number of octets. The last word that is not an integral number of octets is not transferred.

#### **RMAF flag (Multicast Address Frame Receive Flag)**

The RMAF flag indicates that a multicast frame was received.

#### **TRO flag (Transmit Retry Over Flag)**

The TRO flag indicates that a collision occurred again during the 15th retry of frame transmission.

#### **CD flag (Late Collision Detect Flag)**

The CD flag indicates that a late collision was detected during frame transmission.

#### **DLC flag (Loss of Carrier Detect Flag)**

The DLC flag indicates that a loss of carrier was detected during frame transmission.

#### **CND flag (Carrier Not Detect Flag)**

The CND flag sets to 1 when a carrier is not detected during preamble transmission.

#### **RFOF flag (Receive FIFO Overflow Flag)**

The RFOF flag indicates that the receive FIFO overflowed during frame reception.

#### **RDE flag (Receive Descriptor Empty Flag)**

The RDE flag indicates that the read receive descriptor is invalid. When this flag sets to 1, set the RD0.RACT bit in the receive descriptor to 1 and set the EDRRR.RR bit to 1 to resume reception.

#### **FR flag (Frame Receive Flag)**

The FR flag indicates that a frame was received and the receive descriptor was updated. The FR flag sets to 1 every time a frame is received.

#### **TFUF flag (Transmit FIFO Underflow Flag)**

The TFUF flag indicates that no data remains in the transmit FIFO during frame transmission. Incomplete data is sent to the line.

**TDE flag (Transmit Descriptor Empty Flag)**

The TDE flag indicates that the TD0.TACT bit of the transmit descriptor is 0 while the previous transmit descriptor indicates that the frame is not complete (TD0.TFP[1:0] bits are 10b or 00b) in multi-buffer frame transmission. As a result, an incomplete frame might be sent.

When this flag sets to 1, perform a software reset and then set the EDTRR.TR bit to 1 to resume transmission. Transmission starts from the address stored in the TDLAR register.

**TC flag (Frame Transfer Complete Flag)**

The TC flag indicates that all the data specified in the transmit descriptor was transmitted from the ETHERC. This flag is set to 1 when one frame was transmitted in a single-buffer frame transmission or when the last data of a frame is transmitted in multi-buffer frame transmission and the TD0.TACT bit in the next transmit descriptor is 0. After frame transmission is complete, the EDMAC writes the transfer status back to the descriptor.

**ECI flag (ETHERC Status Register Source Flag)**

The ECI flag is set to 1 when an interrupt request is generated by the ECSR register.

**ADE flag (Address Error Flag)**

The ADE flag indicates that the memory address that the EDMAC tried to use for transfer is invalid.

**RFCOF flag (Receive Frame Counter Overflow Flag)**

The RFCOF flag indicates that the next frame reception started while the number of frames stored in the receive FIFO reached the maximum number of frames (16 frames). The received frame is discarded while the RFCOF flag is 1.

**RABT flag (Receive Abort Detect Flag)**

The RABT flag indicates that the ETHERC aborted frame reception because of a CRC error, PHY-LSI receive error, frame-too-short error, frame-too-long error, or other errors.

**TABT flag (Transmit Abort Detect Flag)**

The TABT flag indicates that the ETHERC aborted frame transmission because of transmit retry over, loss of carrier, no carrier detection, or other errors.

**TWB flag (Write-Back Complete Flag)**

The TWB flag indicates the EDMAC completed writing back to the descriptor after frame transmission. This flag is set to 1 after each frame transmission when the TRIMD.TIM bit is 0. It only sets to 1 when the TRIMD.TIS bit is 1.

**28.2.7 EESIPR : ETHERC/EDMAC Status Interrupt Enable Register**

Base address: EDMAC0 = 0x4035\_4000  
EDMAC0\_NS = 0x5035\_4000

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	TWBIP	—	—	—	TABTI P	RABTI P	RFCO FIP	ADEIP	ECIIP	TCIP	TDEIP	TFUFI P	FRIP	RDEIP	RFOFI P
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CNDIP	DLCIP	CDIP	TROIP	RMAFI P	—	—	RRFIP	RTLFI P	RTSFI P	PREIP	CERFI P
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CERFIP	CRC Error Interrupt Request Enable 0: Disable CRC error interrupt requests 1: Enable CRC error interrupt requests.	R/W

Bit	Symbol	Function	R/W
1	PREIP	PHY-LSI Receive Error Interrupt Request Enable 0: Disable PHY-LSI receive error interrupt requests 1: Enable PHY-LSI receive error interrupt requests.	R/W
2	RTSFIP	Frame-Too-Short Error Interrupt Request Enable 0: Disable frame-too-short error interrupt requests 1: Enable frame-too-short error interrupt requests.	R/W
3	RTLFIIP	Frame-Too-Long Error Interrupt Request Enable 0: Disable frame-too-long error interrupt requests 1: Enable frame-too-long error interrupt requests.	R/W
4	RRFIIP	Alignment Error Interrupt Request Enable 0: Disable alignment error interrupt requests 1: Enable alignment error interrupt requests.	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	RMAFIIP	Multicast Address Frame Receive Interrupt Request Enable 0: Disable multicast address frame receive interrupt requests 1: Enable multicast address frame receive interrupt requests.	R/W
8	TROIIP	Transmit Retry Over Interrupt Request Enable 0: Disable transmit retry over interrupt requests 1: Enable transmit retry over interrupt requests.	R/W
9	CDIIP	Late Collision Detect Interrupt Request Enable 0: Disable late collision detected interrupt requests 1: Enable late collision detected interrupt requests.	R/W
10	DLCIIP	Loss of Carrier Detect Interrupt Request Enable 0: Disable loss of carrier detected interrupt requests 1: Enable loss of carrier detected interrupt requests.	R/W
11	CNDIIP	Carrier Not Detect Interrupt Request Enable 0: Disable carrier not detected interrupt requests 1: Enable carrier not detected interrupt requests.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	RFOFIIP	Receive FIFO Overflow Interrupt Request Enable 0: Disable overflow interrupt requests 1: Enable overflow interrupt requests.	R/W
17	RDEIIP	Receive Descriptor Empty Interrupt Request Enable 0: Disable receive descriptor empty interrupt requests 1: Enable receive descriptor empty interrupt requests.	R/W
18	FRIIP	Frame Receive Interrupt Request Enable 0: Disable frame reception interrupt requests 1: Enable frame reception interrupt requests.	R/W
19	TFUFIIP	Transmit FIFO Underflow Interrupt Request Enable 0: Disable underflow interrupt requests 1: Enable underflow interrupt requests.	R/W
20	TDEIIP	Transmit Descriptor Empty Interrupt Request Enable 0: Disable transmit descriptor empty interrupt requests 1: Enable transmit descriptor empty interrupt requests.	R/W
21	TCIIP	Frame Transfer Complete Interrupt Request Enable 0: Disable frame transmission complete interrupt requests 1: Enable frame transmission complete interrupt requests.	R/W
22	ECIIP	ETHERC Status Register Source Interrupt Request Enable 0: Disable ETHERC status interrupt requests 1: Enable ETHERC status interrupt requests.	R/W
23	ADEIIP	Address Error Interrupt Request Enable 0: Disable address error interrupt requests 1: Enable address error interrupt requests.	R/W

Bit	Symbol	Function	R/W
24	RFCOFIP	Receive Frame Counter Overflow Interrupt Request Enable 0: Disable receive frame counter overflow interrupt requests 1: Enable receive frame counter overflow interrupt requests.	R/W
25	RABTIP	Receive Abort Detect Interrupt Request Enable 0: Disable receive abort detected interrupt requests 1: Enable receive abort detected interrupt requests.	R/W
26	TABTIP	Transmit Abort Detect Interrupt Request Enable 0: Disable transmit abort detected interrupt requests 1: Enable transmit abort detected interrupt requests.	R/W
29:27	—	These bits are read as 0. The write value should be 0.	R/W
30	TWBIP	Write-Back Complete Interrupt Request Enable 0: Disable write-back complete interrupt requests 1: Enable write-back complete interrupt requests.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The EESIPR register enables interrupt requests associated with bits in the EESR register. When a bit in this register is 1, the associated interrupt request is enabled.

### 28.2.8 TRSCER : ETHERC/EDMAC Transmit/Receive Status Copy Enable Register

Base address: EDMAC0 = 0x4035\_4000  
EDMAC0\_NS = 0x5035\_4000

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	RMAF CE	—	—	RRFC E	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	RRFCE	RRF Flag Copy Enable 0: Reflect the EESR.RRF flag status in the RD0.RFE bit of the receive descriptor 1: Do not reflect the EESR.RRF flag status in the RD0.RFE bit of the receive descriptor.	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	RMAFCE	RMAF Flag Copy Enable 0: Reflect the EESR.RMAF flag status in the RD0.RFE bit of the receive descriptor 1: Do not reflect the EESR.RMAF flag status in the RD0.RFE bit of the receive descriptor.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The TRSCER register selects whether the receive status indicated in the EESR.RMAF and RRF flags is reflected in the RFE bit of the receive descriptor as a summary. The bits in this register are associated with bits in the EESR register that have the same number. When the RMAFCE or RRFCE bit is set to 0, the associated receive status is reflected in the RFE bit. When the RMAFCE or RRFCE bit is set to 1, the associated receive status is not reflected.

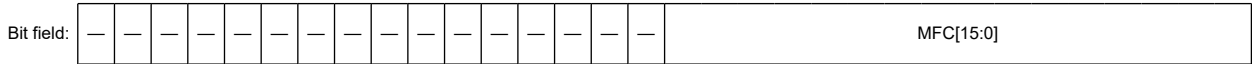


### 28.2.9 RMFCR : Missed-Frame Counter Register

Base address: EDMAC0 = 0x4035\_4000  
 EDMAC0\_NS = 0x5035\_4000

Offset address: 0x40

Bit position: 31 15 0



Value after reset: 0

Bit	Symbol	Function	R/W
15:0	MFC[15:0]	Missed-Frame Counter These bits indicate the number of frames that are discarded and not transferred to the receive buffer during reception.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RMFCR register indicates that the number of frames that could not be stored in the receive FIFO and so were discarded during reception. When the receive FIFO overflows, it stops receiving data, and the rest of frames are discarded. At the same time, the RMFCR register value is incremented. When the RMFCR register value reaches 0xFFFF, count-up is halted. Writing any value to the RMFCR register clears the counter value to 0.

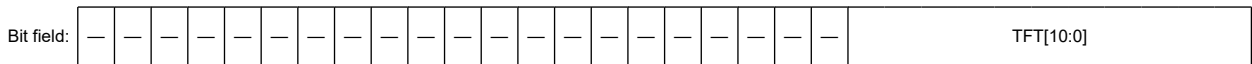
For frames that are not completely received, after data in the receive FIFO is transferred to the receive buffer, the RACT bit in the receive descriptor 0 (RD0) clears to 0 (descriptor disabled), the RFS9 bit sets to 1 (receive FIFO overflowed), and the EESR.RFOF flag is set to 1 (overflow detected).

### 28.2.10 TFTR : Transmit FIFO Threshold Register

Base address: EDMAC0 = 0x4035\_4000  
 EDMAC0\_NS = 0x5035\_4000

Offset address: 0x48

Bit position: 31 10 0



Value after reset: 0

Bit	Symbol	Function	R/W
10:0	TFT[10:0]	Transmit FIFO Threshold 0x000: Store-and-forward mode 0x001 to 0x00C: Setting prohibited 0x00D to 0x200: The threshold is the set value multiplied by 4. 0x201 to 0x7FF: Setting prohibited Example: TFT[10:0] = 0x00D : 52 bytes TFT[10:0] = 0x040 : 256 bytes TFT[10:0] = 0x100 : 1024 bytes TFT[10:0] = 0x200 : 2048 bytes	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Note: When starting transmission before one frame data is completely written, take care to prevent an underflow. To prevent a transmit underflow, Renesas recommends using the initial value (store-and-forward mode).

The TFTR register specifies the transmit FIFO threshold at which the first transmission starts. The actual threshold is the set value multiplied by 4.

The ETHERC starts transmission when the amount of data in the transmit FIFO exceeds the number of bytes set in this register, when the transmit FIFO is full, or when one frame of data is completely written. Set the TFTR register while the EDTRR.TR bit is 0.

### 28.2.11 FDR : FIFO Depth Register

Base address: EDMAC0 = 0x4035\_4000  
 EDMAC0\_NS = 0x5035\_4000

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TFD[4:0]				—	—	—	RFD[4:0]				—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	RFD[4:0]	Receive FIFO Depth 0x0F: 4096 bytes Others: settings prohibited	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
12:8	TFD[4:0]	Transmit FIFO Depth 0x07: 2048 bytes Others: settings prohibited	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The FDR register specifies the transmit and receive FIFO depths. Set this register to 0x0000\_070F before starting transmission and reception.

### 28.2.12 RMCR : Receive Method Control Register

Base address: EDMAC0 = 0x4035\_4000  
 EDMAC0\_NS = 0x5035\_4000

Offset address: 0x58

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

Bit	Symbol	Function	R/W
0	RNR	Receive Request Reset 0: EDRRR.RR bit (receive request bit) is cleared to 0 when one frame is received 1: EDRRR.RR bit (receive request bit) is not cleared to 0 when one frame is received.	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

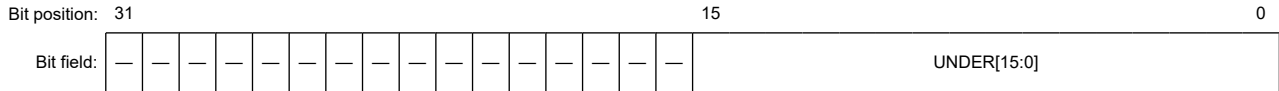
The RMCR register specifies how to control the EDRRR.RR bit when receiving a frame. When the RNR bit is 0, the EDRRR.RR bit clears to 0 when one frame is received, so it must be set to 1 by software to receive the subsequent frame. When the RNR bit is 1, the EDRRR.RR bit does not clear to 0 when one frame is received, and the EDMAC reads the

next receive descriptor and continues frame reception. Renesas recommends setting the RNR bit to 1 when receiving data continuously. Set the RMCR register while the EDRRR.RR bit is 0.

### 28.2.13 TFUCR : Transmit FIFO Underflow Counter

Base address: EDMAC0 = 0x4035\_4000  
EDMAC0\_NS = 0x5035\_4000

Offset address: 0x64



Value after reset: 0

Bit	Symbol	Function	R/W
15:0	UNDER[15:0]	Transmit FIFO Underflow Count These bits indicate how many times the transmit FIFO underflows. The counter stops when the counter value reaches 0xFFFF.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

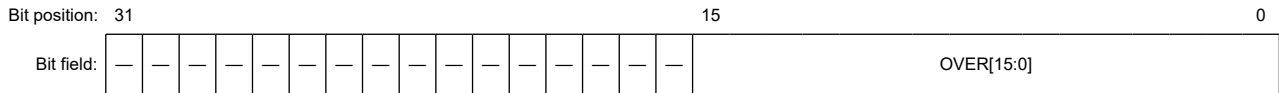
Note: S-TYPE3, P-TYPE3

The TFUCR register indicates how many times the transmit FIFO underflows. Writing any value to the TFUCR register clears the counter value to 0.

### 28.2.14 RFOCR : Receive FIFO Overflow Counter

Base address: EDMAC0 = 0x4035\_4000  
EDMAC0\_NS = 0x5035\_4000

Offset address: 0x68



Value after reset: 0

Bit	Symbol	Function	R/W
15:0	OVER[15:0]	Receive FIFO Overflow Count These bits indicate how many times the receive FIFO overflows. The counter stops when the counter value reaches 0xFFFF.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The RFOCR register indicates how many times the receive FIFO overflows. Writing any value to the RFOCR register clears the counter value to 0.





Bit	Symbol	Function	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

The TRIMD register specifies the transmit interrupt mode and enables or disables transmit interrupts. When the condition selected in this register is satisfied, the EESR.TWB flag sets to 1, and an interrupt request is output when the EESIPR.TWBIP bit is 1.

### 28.2.19 RBWAR : Receive Buffer Write Address Register

Base address: EDMAC0 = 0x4035\_4000  
EDMAC0\_NS = 0x5035\_4000

Offset address: 0xC8

Bit position: 31 0

Bit field:

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	The RBWAR register indicates the last address that the EDMAC wrote data to, when writing to the receive buffer.	R

Note: S-TYPE3, P-TYPE3

The RBWAR register indicates the last address that the EDMAC wrote data to, when writing to the receive buffer. Check the contents of this register to identify the address in the receive buffer to which the EDMAC is writing data to. The address that the EDMAC is outputting to the receive buffer might not match the read value of the RBWAR register during data reception. The RBWAR register is read-only. Do not write to this register.

### 28.2.20 RDFAR : Receive Descriptor Fetch Address Register

Base address: EDMAC0 = 0x4035\_4000  
EDMAC0\_NS = 0x5035\_4000

Offset address: 0xCC

Bit position: 31 0

Bit field:

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	The RDFAR register indicates the start address of the last fetched receive descriptor when the EDMAC is fetching descriptor information from the receive descriptor.	R

Note: S-TYPE3, P-TYPE3

The RDFAR register indicates the start address of the last fetched receive descriptor when the EDMAC is fetching descriptor information from the receive descriptor. Check the contents of this register to identify which receive descriptor information the EDMAC is using for active processing. The address of the receive descriptor that the EDMAC is fetching might not match the read value of the RDFAR register during data reception. The RDFAR register is read-only.

Do not write to this register.

### 28.2.21 TBRAR : Transmit Buffer Read Address Register

Base address: EDMAC0 = 0x4035\_4000  
 EDMAC0\_NS = 0x5035\_4000

Offset address: 0xD4

Bit position: 31 0

Bit field:

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	The TBRAR register indicates the last address from which the EDMAC read data, when reading data from the transmit buffer.	R

Note: S-TYPE3, P-TYPE3

The TBRAR register indicates the last address from which the EDMAC read data, when reading data from the transmit buffer. Check the contents of this register to identify which address in the transmit buffer the EDMAC is reading from.

The address that the EDMAC is outputting to the transmit buffer might not match the read value of the TBRAR register.

The TBRAR register is read-only. Do not write to this register.

### 28.2.22 TDFAR : Transmit Descriptor Fetch Address Register

Base address: EDMAC0 = 0x4035\_4000  
 EDMAC0\_NS = 0x5035\_4000

Offset address: 0xD8

Bit position: 31 0

Bit field:

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	The TDFAR register indicates the start address of the last fetched transmit descriptor when the EDMAC is fetching the descriptor information from the transmit descriptor.	R

Note: S-TYPE3, P-TYPE3

The TDFAR register indicates the start address of the last fetched transmit descriptor when the EDMAC is fetching the descriptor information from the transmit descriptor. Check the contents of this register to identify which transmit descriptor information the EDMAC is using for active processing. The address of transmit descriptor that the EDMAC fetches might not match the read value of the TDFAR register. The TDFAR is read only. Do not write to this register.

## 28.3 Operation

The EDMAC transfers data according to the information written in the descriptor. Two descriptors are provided: transmit and receive. A descriptor includes the buffer size, address, and transmit or receive status. The EDMAC transmits or receives data continuously by using sequentially arranged descriptors.

### 28.3.1 Descriptor Lists and Data Buffers

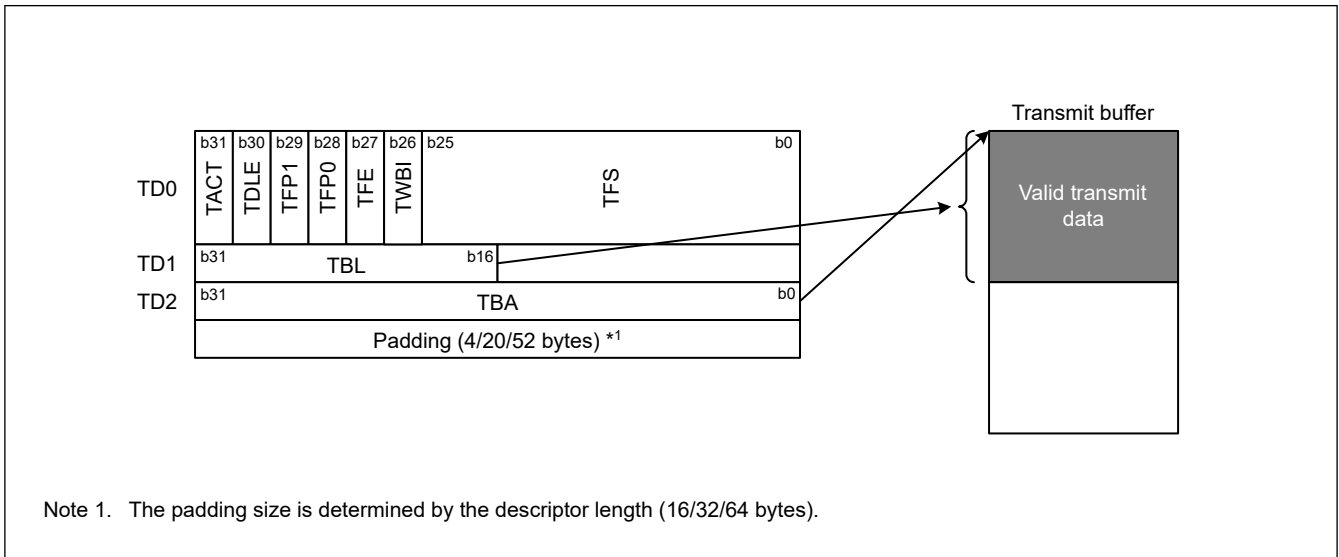
To transfer data using the EDMAC, create the transmit and receive descriptor lists in memory, set the start address of the transmit descriptor list in the TDLAR register, and set the start address of the receive descriptor list in the RDLAR register. Also, the transmit and receive buffers associated with each descriptor are required.

Align the descriptor list on the appropriate address boundary according to the descriptor length set in the EDMR.DL[1:0] bits. The transmit buffer can be aligned on a word boundary, halfword boundary, or byte boundary. When the valid transmit buffer size is 16 bytes or less, align it on a 32-byte boundary. When the valid transmit buffer size is larger than 16bytes, the transmit buffer setting on not aligned 32-byte boundary is permitted. However EDMAC0 might read the transmit buffer aligned on 32-byte boundary, therefore, initialize the transmit buffer area at the beginning of 32 byte-boundary where the

transmit buffer set in. Align the receive buffer on a 32-byte boundary. Set different addresses for the transmit and receive descriptors and buffers for EDMAC0.

### 28.3.1.1 Transmit descriptor

Figure 28.3 shows the relationship between a transmit descriptor and transmit buffer. A transmit descriptor consists of TD0 to TD2. The transmit frame and transmit buffer configuration can be specified as one buffer per frame (single-buffer frame transmission) or multiple buffers per frame (multi-buffer frame transmission) by setting the transmit descriptor.



**Figure 28.3 Relationship between transmit descriptor and transmit buffer**

#### (1) Transmit descriptor 0 (TD0)

Note: Bits for write-back are underlined.

Bit	Symbol	Function	R/W
<u>25:0</u>	<u>TFS</u>	Transmit Frame Status Set all bits to 0 when creating a descriptor. After write-back, the bits indicate the following: TFS25 to TFS9: Reserved TFS8: Transmit abort was detected (value is equivalent to the EESR.TABT flag) TFS7 to TFS4: Reserved TFS3: No carrier was detected (value is equivalent to the EESR.CND flag) TFS2: Loss of carrier was detected (value is equivalent to the EESR.DLC flag) TFS1: Late collision during transmission was detected (value is equivalent to the EESR.CD flag) TFS0: Transmit retry over (value is equivalent to the EESR.TRO flag). When a bit sets to 1, it indicates that the associated error occurred during frame transmission. When any of the TFS bits sets to 1, the TFE bit also is set to 1. When any of bits TFS3 to TFS0 is set to 1, TFS8 is also set to 1.	R/W
26	TWBI	Write-Back Complete Interrupt Enable 0: Do not generate interrupt when write-back to this descriptor is complete 1: Generate interrupt when write-back to this descriptor is complete.	R/W
<u>27</u>	<u>TFE</u>	Transmit Frame Error 0: Frame transmission is successfully complete 1: Error occurred during frame transmission (transmission aborted).	R/W
29:28	TFP[1:0]	Transmit Frame Position 0 0: Transmit buffer indicated in this descriptor is the middle of a transmit frame (frame information is incomplete) 0 1: Transmit buffer indicated in this descriptor is the end of a transmit frame (frame information is complete) 1 0: Transmit buffer indicated in this descriptor is the head of a transmit frame (frame information is incomplete) 1 1: Transmit buffer indicated in this descriptor is all of a transmit frame (one buffer per frame).	R/W



Note: Bits for write-back are underlined.

Bit	Symbol	Function	R/W
30	TDLE	Transmit Descriptor List End When this bit is 1, it indicates that this descriptor is the last in the descriptor list.	R/W
<u>31</u>	<u>TACT</u>	Transmit Descriptor Valid This bit indicates that this descriptor is valid.	R/W

TD0 specifies the transmit frame settings and indicates the status after transmission.

### **TFE bit (Transmit Frame Error)**

When the TFE bit is 1, it indicates that any of the TFS bits is 1.

### **TFP[1:0] bits (Transmit Frame Position)**

The TFP[1:0] bits indicate which part of a transmit frame corresponds to the transmit buffer indicated in this descriptor.

The TFP[1:0] and TD1.TBL bit settings must be logically consistent in the previous and next descriptors.

### **TACT bit (Transmit Descriptor Valid)**

The TACT bit indicates that this descriptor is valid. The TACT bit is set to 1 by software. This bit clears to 0 when the transmit frame is transferred or when the transmission is aborted.

## (2) Transmit descriptor 1 (TD1)

Bit	Symbol	Function	R/W
15:0	—	The read value is 0. The write value should be 0.	R/W
31:16	TBL	Transmit Buffer Length Specifies the valid byte length of the associated transmit buffer. Set a value equal to or greater than 1.	R/W

TD1 specifies the valid byte length of the transmit buffer.

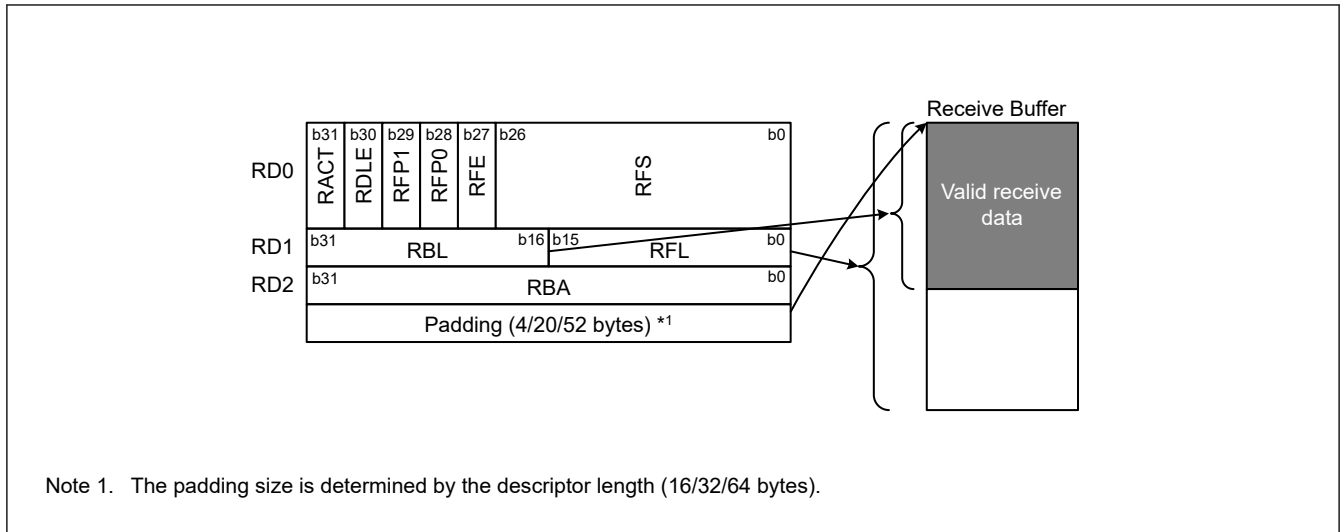
## (3) Transmit descriptor 2 (TD2)

Bit	Symbol	Function	R/W
31:0	TBA	Transmit Buffer Address Specifies the start address of the transmit buffer. When the TD1.TBL bit value is 1 to 16 bytes, align it on a 32-byte boundary.	R/W

TD2 specifies the start address of the transmit buffer.

### 28.3.1.2 Receive descriptor

[Figure 28.4](#) shows the relationship between a receive descriptor and receive buffer. The receive frame and receive buffer configuration can be specified as one buffer per frame (single-buffer frame transmission) or multiple buffers per frame (multi-buffer frame transmission) by setting the receive descriptor. If the receive buffer length (RBL) is set to 0, the operation indicated in the descriptor is not guaranteed.



Note 1. The padding size is determined by the descriptor length (16/32/64 bytes).

**Figure 28.4 Relationship between receive descriptor and receive buffer**

(1) Receive descriptor 0 (RD0)

Note: Bits for write-back are underlined.

Bit	Symbol	Function	R/W
<u>26:0</u>	<u>RFS</u>	<p>Receive Frame Status</p> <p>Set all bits to 0 when creating a descriptor. After write-back, the bits indicate the following:</p> <p>RFS26 to RFS10: Reserved</p> <p>RFS9: Receive FIFO overflow (value is equivalent to the EESR.RFOF flag)</p> <p>RFS8: Receive abort was detected (value is equivalent to the EESR.RABT flag)</p> <p>RFS7: Multicast address frame was received (value is equivalent to the EESR.RMAF flag)</p> <p>RFS6 and RFS5: Reserved</p> <p>RFS4: Alignment error was detected (value is equivalent to the EESR.RRF flag)</p> <p>RFS3: Frame-too-long error (value is equivalent to the EESR.RTLF flag)</p> <p>RFS2: Frame-too-short error (value is equivalent to the EESR.RTSF flag)</p> <p>RFS1: PHY-LSI receive error (value is equivalent to the EESR.PRE flag)</p> <p>RFS0: CRC error (value is equivalent to the EESR.CERF flag).</p> <p>When a bit sets to 1, it indicates that the associated error occurred during frame reception. When any of the RFS bits sets to 1, the RFE bit also sets to 1. Set the TRSCER register to select whether bits RFS7 and RFS4 are reflected in the RFE bit. When any of bits RFS3 to RFS0 sets to 1, RFS8 also sets to 1.</p>	R/W
<u>27</u>	<u>RFE</u>	<p>Receive Frame Error</p> <p>0: No error occurred in the received frame</p> <p>1: Error occurred in the received frame.</p>	R/W
<u>29:28</u>	<u>RFP[1:0]</u>	<p>Receive Frame Position</p> <p>0 0: Receive buffer indicated in this descriptor is the middle of a receive frame (frame information is incomplete)</p> <p>0 1: Receive buffer indicated in this descriptor is the end of a receive frame (frame information is complete)</p> <p>1 0: Receive buffer indicated in this descriptor is the head of a receive frame (frame information is incomplete)</p> <p>1 1: Receive buffer indicated in this descriptor is all of a receive frame (one buffer per frame).</p>	R/W
30	RDLE	<p>Receive Descriptor List End</p> <p>When this bit is 1, it indicates that this descriptor is the last in the descriptor list.</p>	R/W
<u>31</u>	<u>RACT</u>	<p>Receive Descriptor Valid</p> <p>Indicates that this descriptor is valid.</p>	R/W

RD0 indicates the receive frame status.

**RFE bit (Receive Frame Error)**

When the RFE bit is 1, it indicates that any of the RFS bits is 1. Set the TRSCER register to select whether the RFS7 and RFS4 bits of EDMAC0 are reflected in the RFE bit.

**RFP[1:0] bits (Receive Frame Position)**

The RFP[1:0] bits indicate which part of a receive frame corresponds to the receive buffer indicated in this descriptor.

**RACT bit (Receive Descriptor Valid)**

The RACT bit indicates that this descriptor is valid. The RACT bit is set to 1 by software. This bit clears to 0 when all the data is transferred to the receive buffer indicated in RD2 or when the receive buffer becomes full.

**(2) Receive descriptor 1 (RD1)**

Bit	Symbol	Function	R/W
<u>15:0</u>	<u>RFL</u>	Receive Frame Length Specifies the length (number of bytes) of the receive frame stored in the buffer. This does not include the number of bytes for padding set in the RPADIR register. These bits are written back to the descriptor associated with the end of a frame.	R/W
31:16	RBL	Receive Buffer Length Specifies the byte length of the associated receive buffer. Set an integral multiple of 32 as the buffer length.	R/W

Note: Bits for write-back are underlined.

RD1 specifies the receive buffer length. When reception is complete, the receive frame length is written back.

**(3) Receive descriptor 2 (RD2)**

Bit	Symbol	Function	R/W
31:0	RBA	Receive Buffer Address Specifies the start address of the receive buffer. Align the buffer address on a 32-byte boundary.	R/W

RD2 specifies the start address of the receive buffer.

**28.3.2 Transmission**

When the EDTRR.TR bit is set to 1 while the ETHERC0.ECMR.TE bit is 1, the EDMAC reads the descriptor following the previously used descriptor in the transmit descriptor list (or the descriptor indicated in the TDLAR register after a reset). When the TACT bit is 1 in the transmit descriptor (TD0), the EDMAC sequentially reads transmit data from the start address of the transmit buffer indicated in transmit descriptor 2 (TD2) and transfers it to the ETHERC through the transmit FIFO. The ETHERC creates a transmit frame and starts transmission to the MII or RMII. When all data indicated in the TD1.TBL bit is transferred, write-back is performed based on the TD0.TFP[1:0] bit setting as follows:

- When the TD0.TFP[1:0] bits are 00b or 10b (frame is incomplete), the TD0.TACT bit is written back
- When the TD0.TFP[1:0] bits are 01b or 11b (frame is complete), the TD0.TACT, TD0.TFS, and TD0.TFE bits are written back.

When the TD0.TACT bit in the read descriptor is 1, the EDMAC continues reading descriptors and transmit frames. When the TD0.TACT bit in the read descriptor is 0, the EDMAC sets the EDTRR.TR bit to 0 and stops transmission.

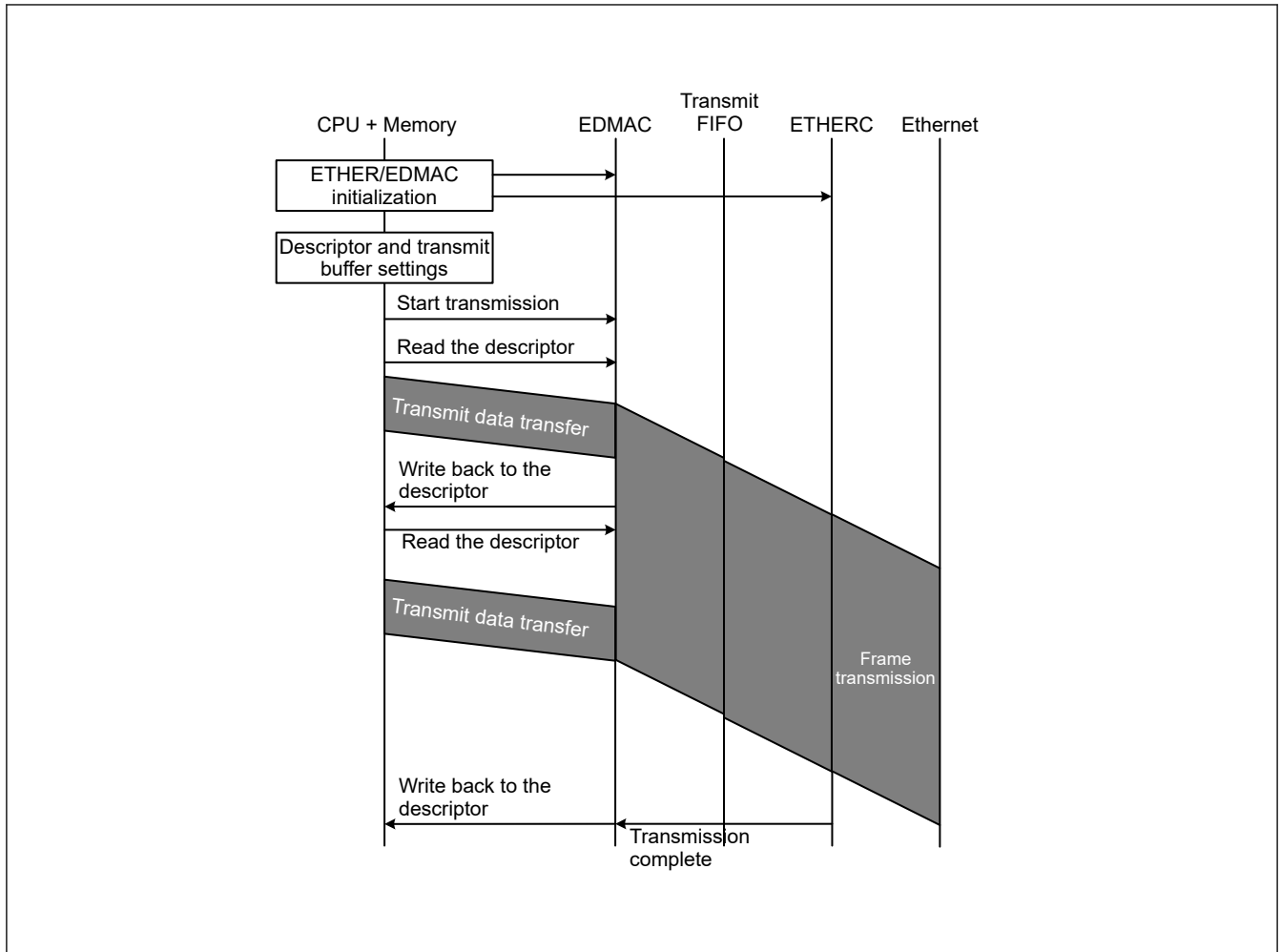


Figure 28.5 Example of transmission flow

### 28.3.3 Reception

When the EDRRR.RR bit is set to 1 while the ETHERC0.ECMR.RE bit is 1, the EDMAC reads the receive descriptor following the previously used descriptor (or the descriptor indicated in the RDLAR register after a reset) and then waits for reception. When the RD0.RACT bit is 1, if the data stored in the receive FIFO is 32 bytes or more, or if the end byte of the frame is stored in the receive buffer, the EDMAC transfers data from the receive FIFO to the receive buffer indicated in receive descriptor 2 (RD2).

If the data length of the received frame is longer than the buffer length set in the RBL bit in receive descriptor 1 (RD1), the EDMAC writes back 10b or 00b to the RD0.RFP[1:0] bits and 0 to the RD0.RACT bit when the receive buffer becomes full, and then the EDMAC reads the next data. After that, the EDMAC transfers data to another receive buffer.

When the frame reception is complete or when the frame reception is aborted by an error, the EDMAC writes back 11b or 01b to the RD0.RFP[1:0] bits, 0 to the RD0.RACT bit, and the receive frame length to the RD1.RFL bit. When the RMCR.RNR bit is 1, the EDMAC reads the next descriptor and waits for reception. When the RNR bit is 0, the EDMAC sets the EDRRR.RR bit to 0 and stops reception.

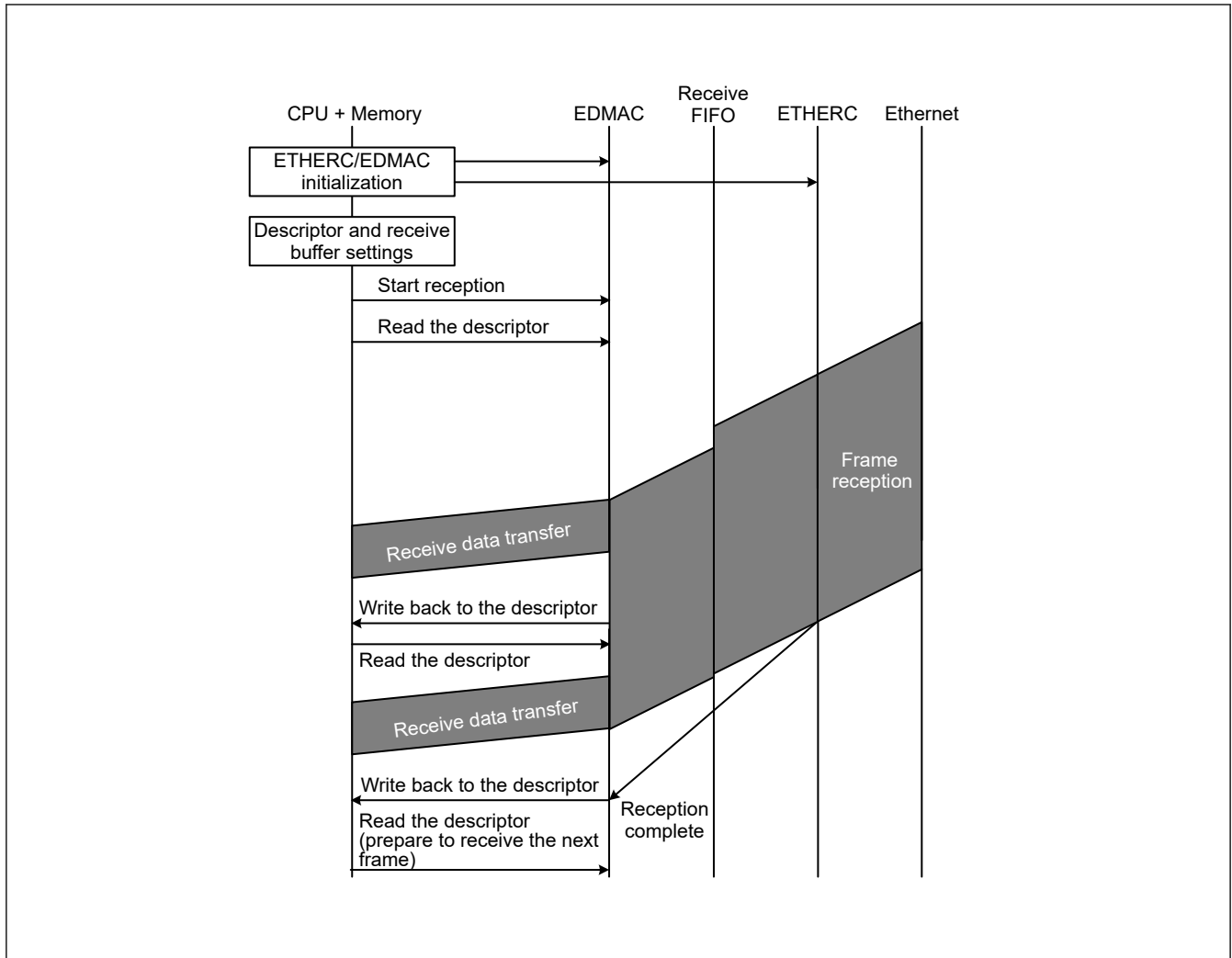


Figure 28.6 Example of reception flow

## 28.3.4 Multi-Buffer Frame Transmission

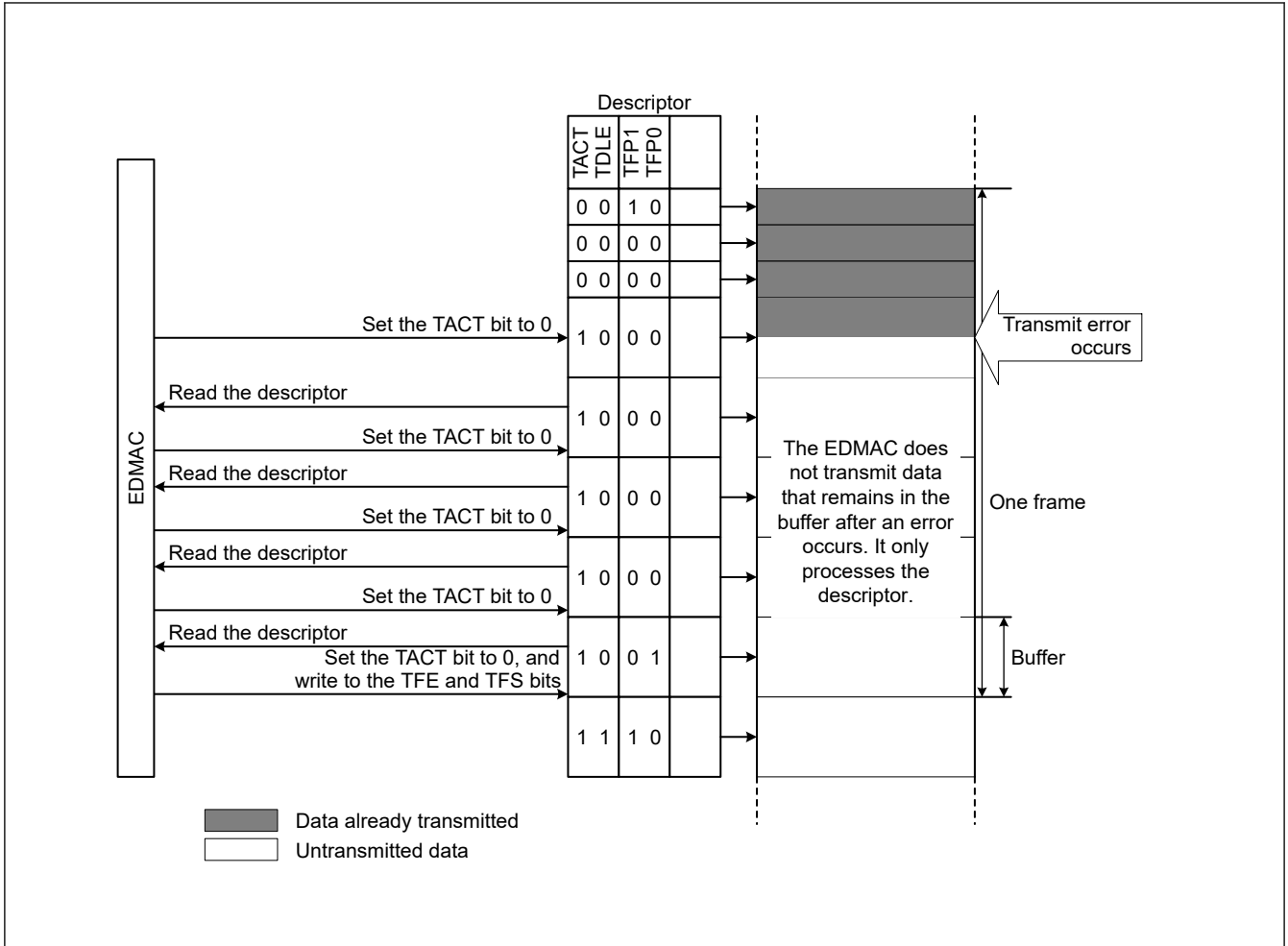
### 28.3.4.1 Error processing while transmitting multi-buffer frame

If an error occurs during multi-buffer frame transmission, the EDMAC performs the processing shown in Figure 28.7. In the figure, when the TACT bit of transmit descriptor 0 (TD0) is 0, the descriptor indicates that all data in the buffer is successfully transmitted. When the TACT bit is 1, the descriptor indicates that data in the buffer is not yet transmitted. If a frame transmit error<sup>\*1</sup> occurs in the head or at the middle of the frame while the TD0.TACT bit is 1, the EDMAC stops data transmission from the transmit FIFO and EDMAC data transfer, and sets the TD0.TACT bit to 0.

After that, the EDMAC reads the next descriptor to see if the descriptor indicates the middle of the frame (TD0.TFP[1:0] bits are 00b) or the end of the frame (TD0.TFP[1:0] bits are 01b). When the descriptor indicates the middle of the frame, the EDMAC sets the TD0.TACT bit to 0 and reads the next descriptor. When the descriptor indicates the end of the frame, in addition to setting the TD0.TACT bit to 0, the EDMAC also writes back to the TD0.TFE and TD0.TFS bits.

After an error occurs, data in the buffer is not transmitted until write-back to the descriptor for the end of the frame. When the associated transmit error interrupt is enabled in the EESIPR register, an interrupt request is generated immediately after write-back to the descriptor for the end of the frame.

Note 1. A transmit retry-over condition, late collision, or loss of carrier is detected, or a carrier is not detected.



**Figure 28.7 EDMAC operation after transmit error occurs**

### 28.3.4.2 Error processing while receiving multi-buffer frame

If an error occurs during multi-buffer frame reception, the EDMAC performs the processing shown in Figure 28.8. In the figure, when the RACT bit of receive descriptor 0 (RD0) is 0, the descriptor indicates that data was successfully received in the buffer. When the RACT bit is 1, the descriptor indicates that data is not yet received in the buffer. If a frame receive error<sup>\*1</sup> occurs, the EDMAC stops receiving new data, but it transfers data that is already stored in the receive FIFO to the receive buffer.

When the receive buffer becomes full during transfer, the EDMAC sets the RACT bit to 0 and the RFP[1:0] bits to 10b or 00b and reads the next descriptor. After all data in the receive FIFO is transferred, the EDMAC writes back the status to the descriptor.

When the associated receive error interrupt is enabled in the EESIPR register, an interrupt request is generated immediately after write-back to the descriptor. When there is a request to receive a new frame, the EDMAC continues reception using the descriptor following the descriptor where the error occurred.

Note 1. A CRC error, PHY-LSI receive error, frame-too-short error, frame-too-long error, or alignment error is detected.

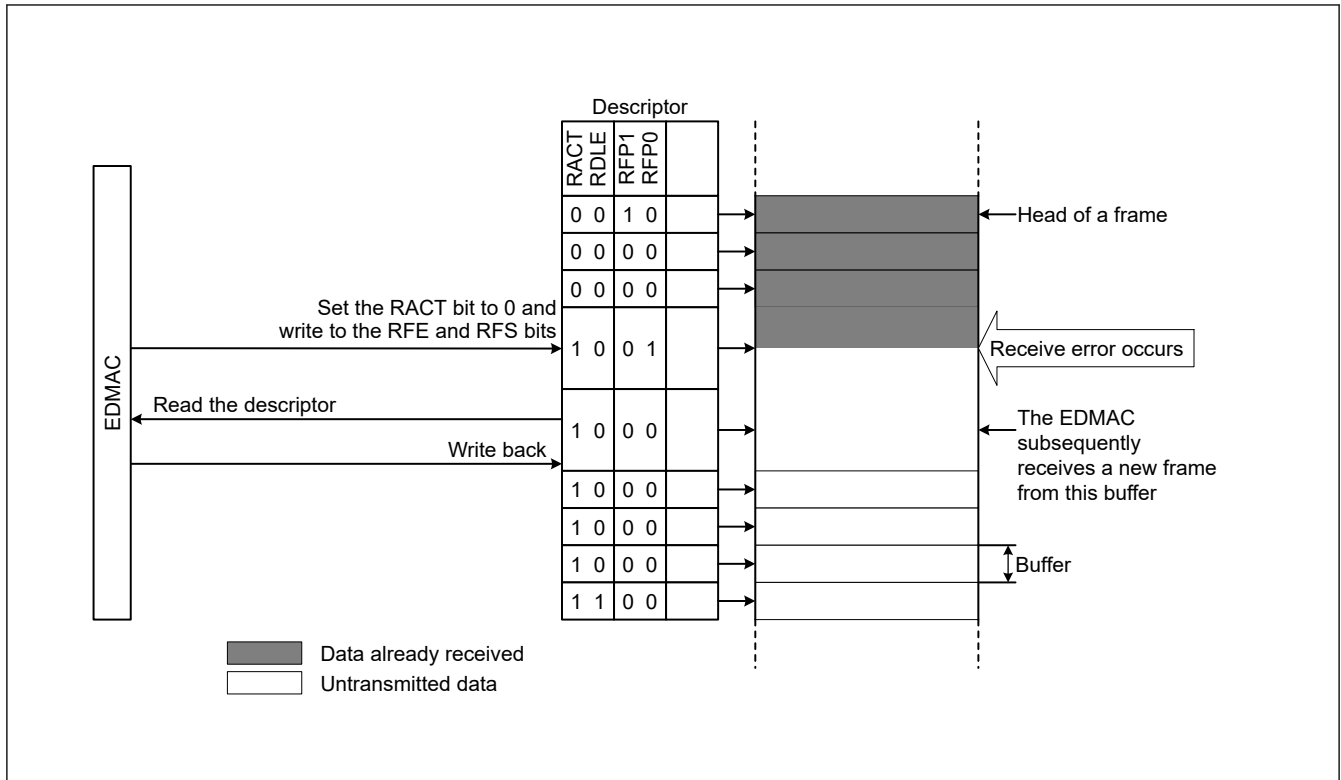


Figure 28.8 EDMAC operation after receive error occurs

### 28.3.5 Bus Transfer Error

Bus transfer error occurs with the slave TrustZone filter error, the master MPU error, the slave bus error or the illegal address access error. When the bus transfer error is detected, the EDMAC halts the process and EESR.ADE bit is set to 1.

Bus transfer error can be output as an interrupt request signal (ETHER\_EINT0). The slave TrustZone error and the master MPU error can be output as NMI. When ETHER\_EINT0 and NMI are generated, NMI always responds first. [Figure 28.9](#) and [Figure 28.10](#) show the processing for the bus error in NMI handler and ETHER\_EINT0 handler.

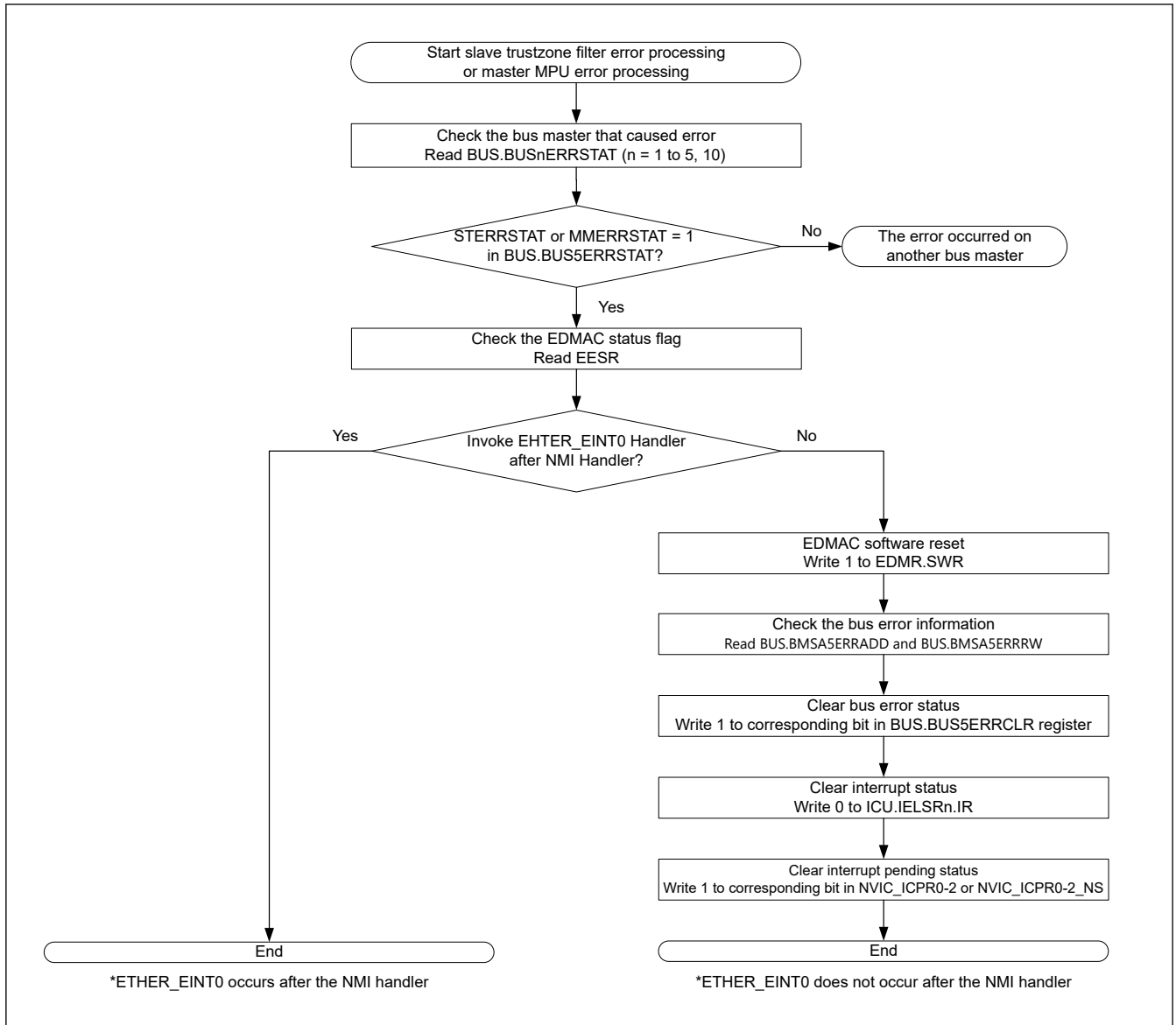
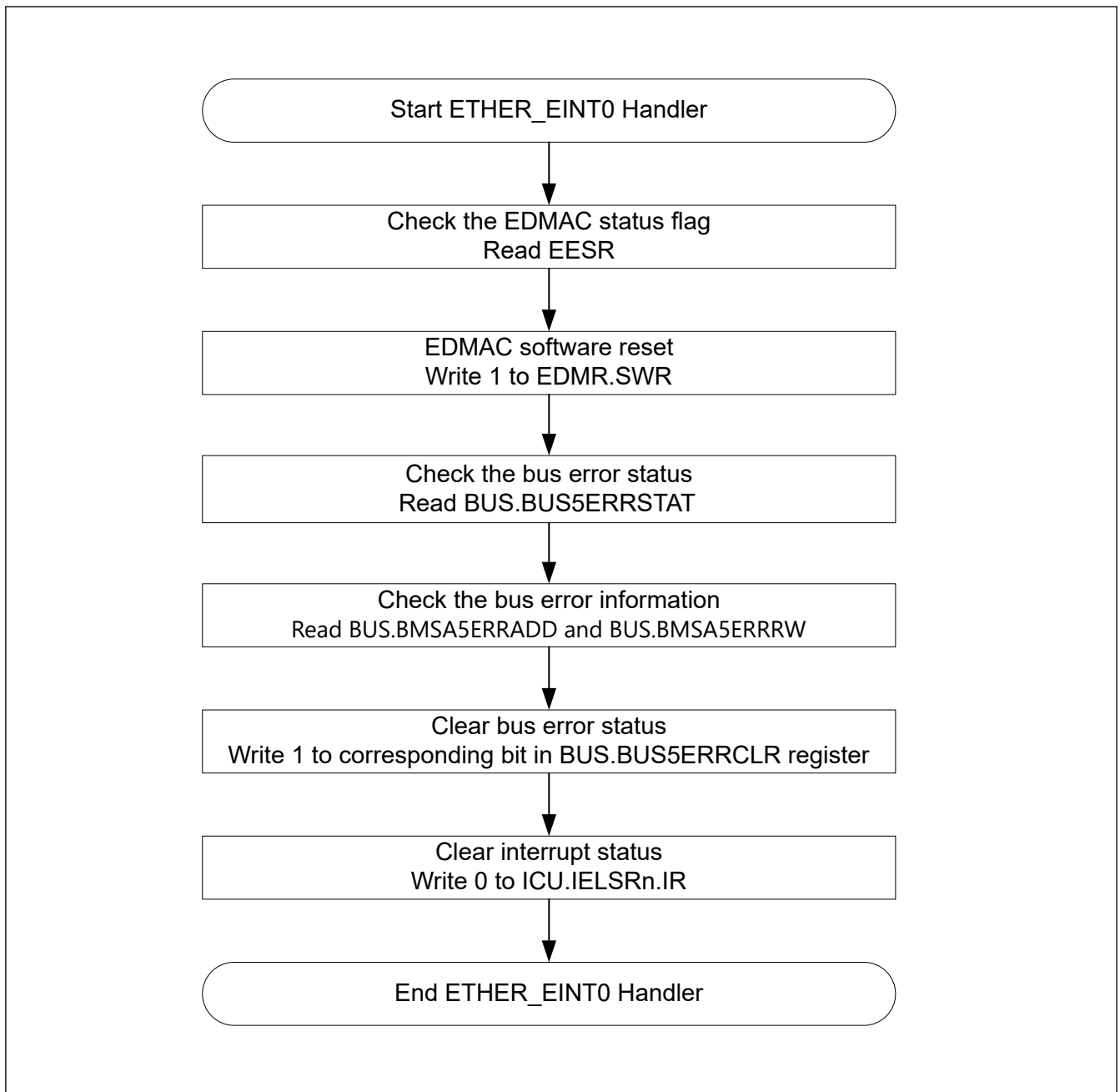


Figure 28.9 Processing for bus transfer error in NMI handler





**Figure 28.10** Processing for bus transfer error in ETHER\_EINT0 handler

## 28.4 Interrupts

When any of the status flags in the EESR register sets to 1 while the associated interrupt request enable bit in the EESIPR register is 1, EDMAC0 issues an ETHER\_EINT0 interrupt request.

## 28.5 Usage Notes

### 28.5.1 Settings for the Module-Stop Function

The following bit in Module Stop Control Register B (MSTPCRB) enables or disables EDMAC module operation:

- The MSTPCRB.MSTPB15 bit in enables or disables ETHERC0 and EDMAC0 operation

The modules are initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

### 28.5.2 Stopping the EDMAC during Operation

When stopping EDMAC operation by using a Software Standby mode or the module-stop function while the EDMAC is running, confirm that the EDTRR.TR and EDRRR.RR bits are 0. If the EDMAC is stopped while the EDTRR.TR or EDRRR.RR bit is 1, data for the frame that is being transmitted or received might not be complete, and EDMAC operation after exiting Software Standby mode or the module-stop state is not guaranteed.

## 29. USB 2.0 Full-Speed Module (USBFS)

### 29.1 Overview

The USB 2.0 Full-Speed module (USBFS) operates as a host or device controller compliant with the Universal Serial Bus (USB) specification revision 2.0. The host controller supports USB 2.0 full-speed and low-speed transfers, and the device controller supports USB 2.0 full-speed transfers. The USBFS has an internal USB transceiver and supports all of the transfer types defined in the USB 2.0 specification

The USBFS has FIFO buffer for data transfers, providing a maximum of 10 pipes. Any endpoint number can be assigned to pipes 1 to 9, based on the peripheral devices or the communication requirements for your system.

Table 29.1 lists the USBFS specifications, Figure 29.1 shows a block diagram, and Table 29.2 lists the I/O pins.

**Table 29.1 USBFS specifications**

Parameter	Specifications
Features	<ul style="list-style-type: none"> <li>• USB Device Controller (UDC) and USB 2.0 transceiver supporting host controller, and device controller, and On-The-Go (OTG) functions</li> <li>• Host and device controller can be switched by software</li> <li>• Self-power or bus power mode selectable</li> </ul>
	Host controller features: <ul style="list-style-type: none"> <li>• Full-speed transfer (12 Mbps) and low-speed transfer (1.5 Mbps)</li> <li>• Automatic scheduling for SOF and packet transmissions</li> <li>• Programmable intervals for isochronous and interrupt transfers</li> <li>• Communications with multiple peripheral devices connected through a single hub</li> </ul>
	Device controller features: <ul style="list-style-type: none"> <li>• Full-speed transfer (12 Mbps)<sup>*1</sup></li> <li>• Control transfer stage control function</li> <li>• Device state control function</li> <li>• Auto response function for SET_ADDRESS request</li> <li>• SOF interpolation</li> </ul>
Supported transfer types	<ul style="list-style-type: none"> <li>• Control transfer</li> <li>• Bulk transfer</li> <li>• Interrupt transfer</li> <li>• Isochronous transfer</li> </ul>
Pipe configuration	<ul style="list-style-type: none"> <li>• FIFO buffer for USB communication</li> <li>• Up to 10 pipes selectable, including the Default Control Pipe (DCP)</li> <li>• Pipes 1 to 9 assignable to any endpoint number</li> </ul>
	Transfer conditions specifiable for each pipe: <ul style="list-style-type: none"> <li>• Pipe 0: Control transfer with 64-byte single buffer</li> <li>• Pipes 1 and 2: Selectable to bulk transfer with 64-byte double buffer or isochronous transfer with 256-byte double buffer</li> <li>• Pipes 3 to 5: Bulk transfer with 64-byte double buffer</li> <li>• Pipes 6 to 9: Interrupt transfer with 64-byte single buffer</li> </ul>
Other features	<ul style="list-style-type: none"> <li>• Reception end function using transaction count</li> <li>• Function that changes the BRDY interrupt event notification timing (BFRE)</li> <li>• Automatic clearing of the FIFO buffer after the data for the pipe specified in the DnFIFO port (n = 0, 1) is read (DCLRM)</li> <li>• NAK setting function for response PID generated on transfer end (SHTNAK)</li> <li>• On-chip pull-up and pull-down resistors for D+ and D-</li> </ul>
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set

Note 1. Low-speed transfer (1.5 Mbps) is not supported.

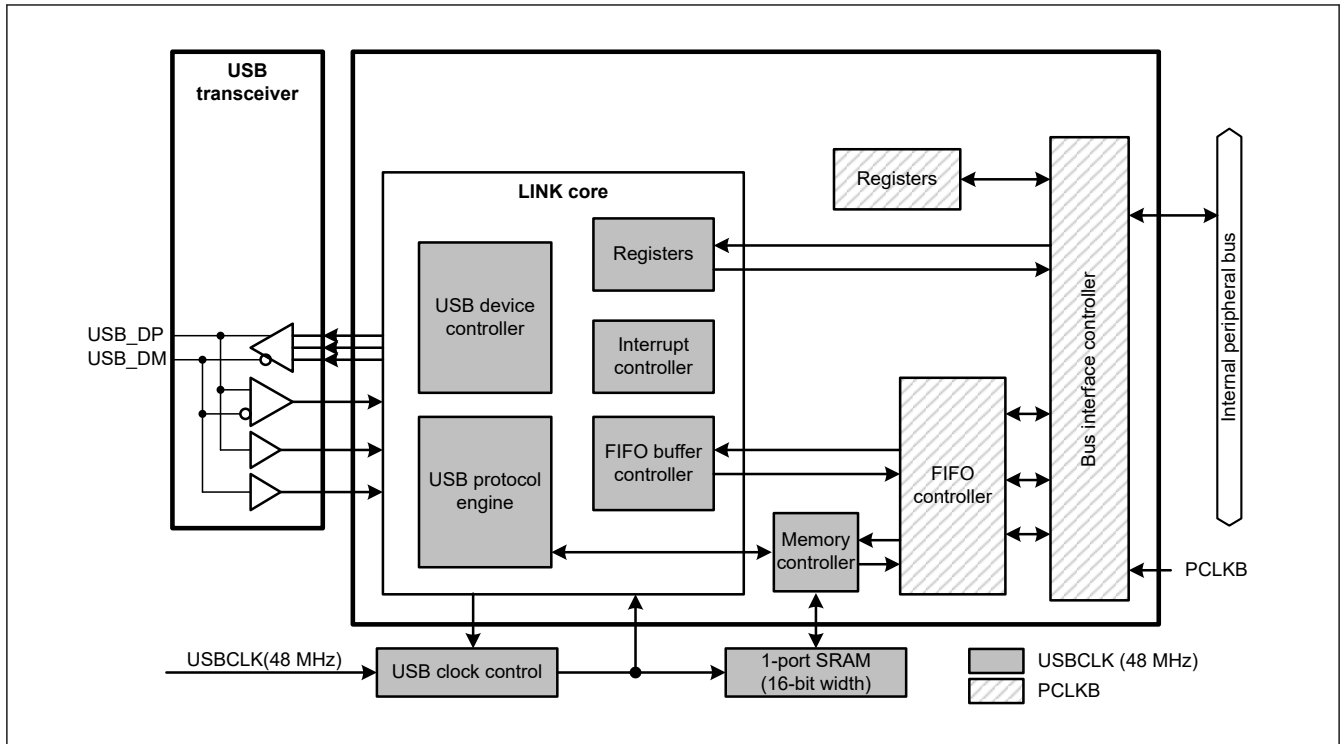


Figure 29.1 USBFS block diagram

Table 29.2 USBFS pin configuration

Function	Pin name	I/O	Description
USBFS	USB_DP	I/O	D+ I/O pin of the USB on-chip transceiver. This pin should be connected to the D+ pin of the USB bus.
	USB_DM	I/O	D- I/O pin of the USB on-chip transceiver. This pin should be connected to the D- pin of the USB bus.
	USB_VBUS	Input	USB cable connection monitor pin. This pin should be connected to VBUS of the USB bus. The VBUS pin status (connected or disconnected) can be detected when the USB module is operating as a function controller.
	USB_EXICEN	Output	Low-power control signal for external power supply (OTG) chip.
	USB_VBUSEN	Output	VBUS (5 V) supply enable signal for external power supply chip.
	USB_OVRCURA, USB_OVRCURB, USB_OVRCURA-DS, USB_OVRCURB-DS	Input	External overcurrent detection signals should be connected to these pins. VBUS comparator signals should be connected to these pins when the OTG power supply chip is connected. USB_OVRCURA or USB_OVRCURB pins can be used in Software standby mode or in normal mode. USB_OVRCURA-DS, USB_OVRCURB-DS are dedicated pins that can generate interrupt in Software standby mode or in normal mode as well as for cancel Deep Software Standby mode 1.
	USB_ID	Input	MicroAB connector ID input signal should be connected to this pin during operation in OTG mode.
	VCC_USB	Input	Power supply pins.
VSS_USB	Input	Ground pins.	

## 29.2 Register Descriptions

### 29.2.1 SYSCFG : System Configuration Control Register

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SCKE	—	—	—	DCFM	DRPD	DPRP U	—	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USBE	USBFS Operation Enable 0: Disable 1: Enable	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	DPRPU	D+ Line Resistor Control 0: Disable line pull-up 1: Enable line pull-up	R/W
5	DRPD	D+/D– Line Resistor Control 0: Disable line pull-down 1: Enable line pull-down	R/W
6	DCFM	Controller Function Select 0: Select device controller 1: Select host controller	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
9:8	—	These bits are read as 0. The write value should be 0.	R/W
10	SCKE	USB Clock Enable 0: Stop clock supply to the USBFS 1: Enable clock supply to the USBFS	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: After writing 1 to the SCKE bit, read it to confirm that it is set to 1.

#### USBE bit (USBFS Operation Enable)

The USBE bit enables or disables operation of the USBFS.

Changing the USBE bit from 1 to 0 initializes the bits listed in [Table 29.3](#). Only change this bit while the SCKE bit is 1. In host controller mode, this bit must be set to 1 after setting the DRPD bit to 1, eliminating SYSSTS0.LNST[1:0] flags chattering, and confirming that the USB bus state is stable.

**Table 29.3 Registers initialized by writing 0 to the SYSCFG.USBE bit**

Selected function	Register	Bit	Remarks
Device controller	SYSSTS0	LNST[1:0]	Value is saved in host controller mode
	DVSTCTR0	RHST[2:0]	—
	INTSTS0	DVSQ[2:0]	Value is saved in host controller mode
	USBADDR	USBADDR[6:0]	Value is saved in host controller mode
	USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]	Value is saved in host controller mode
	USBVAL	WVALUE[15:0]	Value is saved in host controller mode
	USBINDX	WINDEX[15:0]	Value is saved in host controller mode
	USBLENG	WLENTUH[15:0]	Value is saved in host controller mode
Host controller	DVSTCTR0	RHST[2:0]	—
	FRMNUM	FRNM[10:0]	Value is saved in device controller mode

**DPRPU bit (D+ Line Resistor Control)**

The DPRPU bit enables or disables pulling up the D+ line in device controller mode.

When the DPRPU bit is set to 1 in device controller mode, the USBFS pulls up the D+ line to notify the USB host that it attached. Changing the DPRPU bit from 1 to 0 releases the pull-up, thereby notifying the USB host that it detached.

Set this bit to 1 in device controller mode and to 0 in host controller mode.

**DRPD bit (D+/D- Line Resistor Control)**

TheDRPD bit enables or disables pulling down D+ and D- lines in host controller mode.

Set this bit to 1 in host controller mode and to 0 in device controller mode.

**DCFM bit (Controller Function Select)**

The DCFM bit selects the host or device function of the USBFS.

Only change this bit when the DPRPU and DRPD bits are both 0.

**SCKE bit (USB Clock Enable)**

The SCKE bit stops or enables the 48-MHz clock supply to the USBFS.

When this bit is 0, only SYSCFG is permitted to be read from and written to; the other registers related to the USB should not be read from or written to.

**29.2.2 SYSSTS0 : System Configuration Status Register 0**

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x004

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	OVCMON[1:0]	—	—	—	—	—	—	—	HTAC T	SOFE A	—	—	IDMO N	LNST[1:0]
------------	-------------	---	---	---	---	---	---	---	-----------	-----------	---	---	-----------	-----------

Value after reset: x x 0 0 0 0 0 0 0 0 0 0 0 0 x 0 0

Bit	Symbol	Function	R/W
1:0	LNST[1:0]	USB Data Line Status Monitor Indicates the status of the USB data lines, see <a href="#">Table 29.4</a>	R
2	IDMON	External ID0 Input Pin Monitor 0: USB_ID pin is low 1: USB_ID pin is high	R
4:3	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
5	SOFEA	Active Monitor When the Host Controller Is Selected 0: SOF output stopped 1: SOF output operating	R
6	HTACT	USB Host Sequencer Status Monitor 0: Host sequencer completely stopped 1: Host sequencer not completely stopped	R
13:7	—	These bits are read as 0.	R
15:14	OVCMON[1:0]	External USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS Input Pin Monitor OVCMON[1] indicates the USB_OVRCURA or USB_OVRCURA-DS pin status. OVCMON[0] indicates the USB_OVRCURB or USB_OVRCURB-DS pin status.	R

Note: S-TYPE-3, P-TYPE-3

Note: The values of the OVCMON[1:0] and IDMON bits depend on the status of the USB\_OVRCURA, USB\_OVRCURA-DS, USB\_OVRCURB or USB\_OVRCURB-DS and USB\_ID pins.

### LNST[1:0] bits (USB Data Line Status Monitor)

The LNST[1:0] bits indicate the state of the USB data lines, D+ and D-. For details, see [Table 29.4](#).

In device controller mode, read the LNST[1:0] bits after connection processing (SYSCFG.DPRPU bit = 1). In host controller mode, read them after enabling pull-down of the lines (SYSCFG.DRPD bit = 1).

**Table 29.4 Status of the USB data bus lines (D+ and D-)**

LNST[1:0] bits	During full-speed operation	During low-speed operation
00b	SE0	SE0
01b	J-State	K-State
10b	K-State	J-State
11b	SE1	SE1

### SOFEA bit (Active Monitor When the Host Controller Is Selected)

The SOFEA bit is used in host controller mode to check whether the output of the last SOF is complete when the USBFS is suspended because of a 0 setting to the DVSTCTR0.UACT bit.

In host controller mode, check that both the HTACT and SOFEA bits are 0 before setting the SYSCFG.USBE bit to 0 to stop the USBFS or setting the SYSCFG.SCKE bit to 0 to stop the clock signal supply during communication.

### HTACT bit (USB Host Sequencer Status Monitor)

The HTACT bit is set to 0 when the host sequencer of the USBFS is completely stopped.

In host controller mode, check that the HTACT bit is 0 before setting the DVSTCTR0.UACT bit to 0 to place the USBFS in the suspended state or setting the SYSCFG.SCKE bit to 0 to stop the clock signal supply during communication.

### OVCMON[1:0] bits (External USB\_OVRCURA, USB\_OVRCURA-DS, USB\_OVRCURB or USB\_OVRCURB-DS Input Pin Monitor)

The OVCMON[1:0] bits indicate the status of the overcurrent signals from an external power supply IC.

## 29.2.3 DVSTCTR0 : Device State Control Register 0

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	HNPB TOA	EXICE N	VBUS EN	WKUP	RWUP E	USBR ST	RESU ME	UACT	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RHST[2:0]	USB Bus Reset Status 0 0 0: In host controller mode: Communication speed indeterminate (powered state or no connection) In device controller mode: Communication speed indeterminate 0 0 1: In host controller mode: Low-speed connection In device controller mode: USB bus reset in progress 0 1 0: In host controller mode: Full-speed connection In device controller mode: USB bus reset in progress or full-speed connection 0 1 1: Setting prohibited Others: In host controller mode: USB bus reset in progress In device controller mode: Setting prohibited	R
3	—	This bit is read as 0. The write value should be 0.	R/W
4	UACT	USB Bus Enable 0: Disable downstream port (disable SOF transmission) 1: Enable downstream port (enable SOF transmission)	R/W
5	RESUME	Resume Output 0: Do not output resume signal 1: Output resume signal	R/W
6	USBRST	USB Bus Reset Output 0: Do not output USB bus reset signal 1: Output USB bus reset signal	R/W
7	RWUPE	Wakeup Detection Enable 0: Disable downstream port remote wakeup 1: Enable downstream port remote wakeup	R/W
8	WKUP	Wakeup Output 0: Do not output remote wakeup signal 1: Output remote wakeup signal	R/W
9	VBUSEN	USB_VBUSEN Output Pin Control 0: Output low on external USB_VBUSEN pin 1: Output high on external USB_VBUSEN pin	R/W
10	EXICEN	USB_EXICEN Output Pin Control 0: Output low on external USB_EXICEN pin 1: Output high on external USB_EXICEN pin	R/W
11	HNPBTOA	Host Negotiation Protocol (HNP) Control Use this bit when switching from device B to device A in OTG mode. If the HNPBTOA bit is 1, the internal function control remains in the Suspend state until the HNP processing ends even if SYSCFG.DPRPU = 0 or SYSCFG.DCFM = 1.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The USBFS controller does not support low-speed connections in device controller mode. When this value is read, abnormal connection processing must be executed in higher level application software.

### RHST[2:0] bits (USB Bus Reset Status)

RHST[2:0] bits indicate the status of the USB bus reset.

In host controller mode, writing 1 to the USBRST bit causes the RHST[2:0] bits to set to 100b. When 0 is written to the USBRST bit and the USBFS ends the SE0 state, the RHST[2:0]bits update to a new value.

In device controller mode, if the USBFS detects a USB bus reset, the RHST[2:0] bits indicate 010b if the DPRPU bit is 1, and a DVST interrupt is generated.

### UACT bit (USB Bus Enable)

When set to 1 in host controller mode, the UACT bit enables USB bus operation by controlling SOF packet transmission to the USB bus in addition to data and reception. The USBFS starts SOF packet output within one frame period after the UACT bit is set to 1. When UACT is set to 0, the USBFS enters the idle state after the SOF packet output.

The USBFS sets the UACT bit to 0 on any of the following conditions:

- A DTCH interrupt is detected during communication (when UACT = 1)



- An EOFERR interrupt is detected during communication (when UACT = 1)

Always write 1 to the UACT bit at the end of the USB bus reset processing (writing 0 to the USBRST bit) or at the end of resume processing from the suspended state (writing 0 to the RESUME bit).

In device controller mode, always set this bit to 0.

### RESUME bit (Resume Output)

The RESUME bit controls the resume signal output in host controller mode.

When this bit is set to 1, the USBFS drives the USB port to the K-state and outputs the resume signal. The USBFS sets the bit to 1 on detection of a remote wakeup signal while the RWUPE bit is 1 and in the USB Suspend state.

The USBFS continues outputting the K-state while the RESUME bit is 1, until the bit is cleared to 0 by software. The RESUME bit must be 1 (resume period) for the time defined in the USB 2.0 specification. Only set this bit to 1 while the interface is in the Suspend state. Write 1 to the UACT bit simultaneously with the end of the resume processing (writing 0 to the RESUME bit).

Always set this bit to 0 in device controller mode.

### USBRST bit (USB Bus Reset Output)

The USBRST bit controls the output of the USB bus signal in host controller mode. When this bit set to 1, the USBFS drives the USB port to the SE0 state to reset the USB bus. The USBFS continues outputting SE0 while the USBRST bit is 1, until the bit is cleared to 0 by software. The USBRST bit must be 1 (USB bus reset period) for the time defined in the USB 2.0 specification. Writing 1 to the USBRST bit during communication (UACT = 1) or during resume processing (RESUME = 1) prevents the USBFS from starting USB bus reset processing until both the UACT and RESUME bits become 0. Write 1 to the UACT bit simultaneously with the end of the USB bus reset processing (writing 0 to the USBRST bit).

Always set this bit to 0 in device controller mode.

### RWUPE bit (Wakeup Detection Enable)

The RWUPE bit enables or disables remote wakeup signals (resume signals) from downstream peripheral devices in host controller mode. When this bit is set to 1, the USBFS detects a remote wakeup signal (K-state for 2.5  $\mu$ s) from a downstream peripheral device, and performs resume processing, driving the K-state. When the RWUPE bit is set to 0, the USBFS ignores remote wakeup signals (K-states) from peripheral devices connected to the USB port.

Do not stop the internal clock when the RWUPE bit is 1, even in the Suspend state (SYSCFG.SCKE bit must be set to 1).

Always set this bit to 0 in device controller mode.

### WKUP bit (Wakeup Output)

The WKUP bit enables or disables remote wakeup signals (resume signals) to the USB bus in device controller mode.

The USBFS controls the output timing of the remote wakeup signals. When this bit is set to 1, the USBFS clears it to 0 after outputting the K-state for 10 ms. The USB 2.0 specification specifies that the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USBFS writes 1 to the WKUP bit immediately after detecting the Suspend state, the K-state is output after 2 ms.

Only write 1 to the WKUP bit when the device is in the Suspend state (INTSTS0.DVSQL[2:0] = 1xxb) and the USB host enables the remote wakeup signal. Do not stop the internal clock while this bit is 1, even in the Suspend state (SYSCFG.SCKE bit must be set to 1).

Always set this bit to 0 in host controller mode.

### HNPBTOA bit (Host Negotiation Protocol (HNP) Control)

The HNPBTOA bit is used when switching from device B to device A while in OTG mode.

If the HNPBTOA bit is 1, the internal function control maintains the Suspend state until HNP processing ends, even if the SYSCFG.DPRPU bit is set to 0 or the SYSCFG.DCFM bit is set to 1. Resume interrupts (RESM) are not generated even if a falling edge of D+ is detected.

After this bit is set to 1, the HNP processing ends when a host attach event is detected, because of a pull-up by the initiating party, or the HNPBTOA bit is cleared to 0 by software because the HNP processing times out.

### 29.2.4 CFIFO/CFIFOL : CFIFO Port Register

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x014

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field: FIFOPORT[15:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	FIFOPORT[15:0] <sup>*1</sup>	FIFO Port Read receive data from the FIFO buffer or write transmit data to the FIFO buffer by accessing these bits	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. The valid bits depend on the MBW settings (CFIFOSEL.MBW) and BIGEND settings (CFIFOSEL.BIGEND) in the associated port selection register. See [Table 29.5](#) and [Table 29.6](#).

Three FIFO ports are available:

- CFIFO
- D0FIFO
- D1FIFO

Each FIFO port is configured with:

- A port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer and writing of data to the FIFO buffer
- A port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that selects the pipe assigned to the FIFO port
- A port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR)

Each FIFO port has the following constraints:

- Access to the FIFO buffer for DCP control transfers is through the CFIFO port
- Access to the FIFO buffer for DMA or DTC transfers is through the D0FIO or D1FIFO port
- The D0FIFO and D1FIFO ports can also be accessed by the CPU
- When using functions specific to the FIFO port, such as the DMA or DTC transfer function, you cannot change the pipe number selected in the CURPIPE[3:0] bits of the port selection register
- Registers configuring a FIFO port do not affect other FIFO ports
- The same pipe must not be assigned to two or more FIFO ports
- There are two FIFO buffer states, one giving access rights to the CPU and the other to the serial interface engine (SIE). When the SIE has access rights, the FIFO buffer cannot be accessed by the CPU

#### FIFOPORT[15:0] bits (FIFO Port)

When the FIFOPORT[15:0] bit is accessed, the USBFS reads the received data from the FIFO buffer or writes the transmit data to the FIFO buffer. The FIFO port register can be accessed only when the FRDY bit in the associated port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in the FIFO port register depend on the MBW and BIGEND settings in the port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL). See [Table 29.5](#) and [Table 29.6](#).

**Table 29.5 Endian operation in 16-bit access**

CFIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

**Table 29.6 Endian operation in 8-bit access**

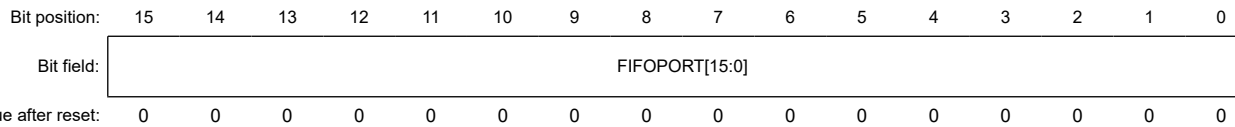
CFIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	Access prohibited*1	N + 0 data
1	Access prohibited*1	N + 0 data

Note 1. Writing to or reading from these areas is not allowed.

### 29.2.5 DnFIFO/DnFIFOL : DnFIFO Port Register (n = 0, 1)

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x018 + 0x4 × n



Bit	Symbol	Function	R/W
15:0	FIFOPORT[15:0]*1	FIFO Port Read receive data from the FIFO buffer or write transmit data to the FIFO buffer by accessing these bits	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. The valid bits depend on the MBW settings (CFIFOSEL.MBW, D0FIFOSEL.MBW, and D1FIFOSEL.MBW) and BIGEND settings (CFIFOSEL.BIGEND, D0FIFOSEL.BIGEND, and D1FIFOSEL.BIGEND) in the associated port selection register. See [Table 29.7](#) and [Table 29.8](#).

Three FIFO ports are available:

- CFIFO
- D0FIFO
- D1FIFO

Each FIFO port is configured with:

- A port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer and writing of data to the FIFO buffer
- A port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that selects the pipe assigned to the FIFO port
- A port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR)

Each FIFO port has the following constraints:

- Access to the FIFO buffer for DCP control transfers is through the CFIFO port
- Access to the FIFO buffer for DMA or DTC transfers is through the D0FIO or D1FIFO port
- The D0FIFO and D1FIFO ports can also be accessed by the CPU
- When using functions specific to the FIFO port, such as the DMA or DTC transfer function, you cannot change the pipe number selected in the CURPIPE[3:0] bits of the port selection register
- Registers configuring a FIFO port do not affect other FIFO ports
- The same pipe must not be assigned to two or more FIFO ports
- There are two FIFO buffer states, one giving access rights to the CPU and the other to the serial interface engine (SIE). When the SIE has access rights, the FIFO buffer cannot be accessed by the CPU

#### FIFOPORT[15:0] bits (FIFO Port)

When the FIFOPORT bit is accessed, the USBFS reads the received data from the FIFO buffer or writes the transmit data to the FIFO buffer. The FIFO port register can be accessed only when the FRDY bit in the associated port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in the FIFO port register depend on the MBW and BIGEND settings in the port selection register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL). See [Table 29.7](#) and [Table 29.8](#).

**Table 29.7 Endian operation in 16-bit access**

CFIFOSEL.BIGEND bit D0FIFOSEL.BIGEND bit D1FIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

**Table 29.8 Endian operation in 8-bit access**

CFIFOSEL.BIGEND bit D0FIFOSEL.BIGEND bit D1FIFOSEL.BIGEND bit	Bits [15:8]	Bits [7:0]
0	Access prohibited*1	N + 0 data
1	Access prohibited*1	N + 0 data

Note 1. Writing to or reading from these areas is not allowed.

### 29.2.6 CFIFOSEL : CFIFO Port Select Register

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x020

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	RCNT	REW	—	—	—	MBW	—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]		
------------	------	-----	---	---	---	-----	---	--------	---	---	------	---	--------------	--	--

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	CFIFO Port Access Pipe Specification 0x0: Default Control Pipe 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	ISEL	CFIFO Port Access Direction When DCP Is Selected 0: Select reading from the FIFO buffer 1: Select writing to the FIFO buffer	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	BIGEND	CFIFO Port Endian Control 0: Little endian 1: Big endian	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	MBW	CFIFO Port Access Bit Width 0: 8-bit width 1: 16-bit width	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
14	REW	Buffer Pointer Rewind 0: Do not rewind buffer pointer 1: Rewind buffer pointer	W*1
15	RCNT	Read Count Mode 0: The DTLN[8:0] bits (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) are cleared when all receive data is read from the CFIFO. In double buffer mode, the DTLN[8:0] value is cleared when all data is read from only a single plane. 1: The DTLN[8:0] bits are decremented each time the receive data is read from the CFIFO.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only 0 can be read.

Do not specify the same pipe number in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected.

Do not change the pipe number while DMA or DTC transfer is enabled.

### CURPIPE[3:0] bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number to use for reading or writing data through the CFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, even when an attempt is made to change the CURPIPE[3:0] setting, the current access setting is retained until access is complete.

### ISEL bit (CFIFO Port Access Direction When DCP Is Selected)

After writing a new value to the ISEL bit with the DCP as the selected pipe, read the ISEL bit to check that the written value agrees with the read value before proceeding to the next process. Set the ISEL and CURPIPE[3:0] bits simultaneously.

### MBW bit (CFIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the CFIFO port.

When the selected pipe is receiving, set the CURPIPE[3:0] and MBW bits simultaneously. After a write to these bits starts a data read from the FIFO buffer, do not change the MBW bit until all of the data is read.

When the selected pipe is transmitting, the bit width cannot be changed from 8-bit to 16-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

### REW bit (Buffer Pointer Rewind)

The REW bit specifies whether to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double buffering, this setting enables re-reading of the currently-read FIFO buffer plane from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

### 29.2.7 DnFIFOSEL : DnFIFO Port Select Register (n = 0, 1)

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x028 + 0x4 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	DCLRM	DREQE	—	MBW	—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	FIFO Port Access Pipe Specification 0x0: Default Control Pipe 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BIGEND	FIFO Port Endian Control 0: Little endian 1: Big endian	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	MBW	FIFO Port Access Bit Width 0: 8-bit width 1: 16-bit width	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	DREQE	DMA/DTC Transfer Request Enable 0: Disable DMA/DTC transfer request 1: Enable DMA/DTC transfer request	R/W
13	DCLRM	Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read 0: Disable auto buffer clear mode 1: Enable auto buffer clear mode	R/W
14	REW	Buffer Pointer Rewind 0: Do not rewind buffer pointer 1: Rewind buffer pointer	W
15	RCNT	Read Count Mode 0: Clear DTLN[8:0] bits in (CFIFOCTR.DTLN[8:0], D0FIFOCTR.DTLN[8:0], D1FIFOCTR.DTLN[8:0]) when all receive data is read from DnFIFO (after read of a single plane in double buffer mode) 1: Decrement DTLN[8:0] bits each time receive data is read from DnFIFO	R/W

Note: S-TYPE-3, P-TYPE-3

The same pipe must not be specified in the CURPIPE[3:0] bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. When the CURPIPE[3:0] bits in the D0FIFOSEL and D1FIFOSEL registers are set to 0000b, no pipe is selected. The pipe number must not be changed while DMA or DTC transfer is enabled.

#### CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number to use for reading or writing data through the DnFIFO port. After writing to these bits, read them to check that the written value agrees with the read value before proceeding to the next process. Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

During FIFO buffer access, even when an attempt is made to change the CURPIPE[3:0] setting, the current access setting is retained until access is complete.

### MBW bit (FIFO Port Access Bit Width)

The MBW bit specifies the bit width for accessing the DnFIFO port.

When the selected pipe is receiving, after a write to these bits starts a data read from the FIFO buffer, do not change the MBW bit until all of the data is read. Set the CURPIPE[3:0] and MBW bits simultaneously.

When the selected pipe is transmitting, the bit width cannot be changed from 8-bit to 16-bit while data is being written to the FIFO buffer.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

### DREQE bit (DMA/DTC Transfer Request Enable)

The DREQE bit enables or disables issuing of DMA or DTC transfer requests. To enable DMA or DTC transfer requests, set this bit to 1 after setting the CURPIPE[3:0] bits. To change the CURPIPE[3:0] setting, first set this bit to 0.

### DCLRM bit (Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read)

The DCLRM bit enables or disables automatic FIFO buffer clearing after data in the selected pipe is read.

When this bit is set to 1, on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or when reading of a received short packet is complete while the PIPECFG.BFRE bit is 1, the USBFS sets the BCLR bit in the FIFO port control register to 1.

When using the USBFS with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

### REW bit (Buffer Pointer Rewind)

The REW bit specifies whether to rewind the buffer pointer.

When the selected pipe is receiving, setting this bit to 1 while the FIFO buffer is being read allows re-reading of the FIFO buffer from the first data. In double buffering, this setting enables re-reading of the currently-read FIFO buffer plane from the first entry.

Do not set this bit to 1 while simultaneously changing the CURPIPE[3:0] bits. Before setting the bit to 1, be sure to check that the FRDY bit is 1.

To rewrite to the FIFO buffer from the first data for the transmitting pipe, use the BCLR bit.

### RCNT bit (Read Count Mode)

The RCNT bit specifies the read mode for the value in the D0FIFOCTL.DTLN bit and D1FIFOCTL.DTLN bit. When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

## 29.2.8 CFIFOCTR : CFIFO Port Control Register

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x022

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	DTLN[8:0]	Receive Data Length Indicates the receive data length. The meaning of the values differs depending on the RCNT bit setting in the port select register. For details, see the description of the DTLN[8:0] bits.	R
12:9	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
13	FRDY	FIFO Port Ready 0: FIFO port access disabled 1: FIFO port access enabled	R
14	BCLR	CPU Buffer Clear 0: No operation 1: Clear FIFO buffer on the CPU side	W
15	BVAL	Buffer Memory Valid Flag 0: Invalid (writing 0 has no effect) 1: Writing ended	R/W

Note: S-TYPE-3, P-TYPE-3

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

### DTLN[8:0] bits (Receive Data Length)

The DTLN[8:0] bits indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the DnFIFOSEL.RCNT bit ( $n = 0, 1$ ), as follows:

- RCNT = 0

The USBFS sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU or DMA/DTC has read all of the received data from a single FIFO buffer plane.

While the PIPECFG.BFRE bit = 1, the USBFS retains the length of the receive data until the BCLR bit is set to 1, even after all the data is read.

- RCNT = 1

The USBFS decrements the value indicated in the DTLN[8:0] bits each time data is read from the FIFO buffer. The value is decremented by 1 when MBW = 0, and by 2 when MBW = 1.

The USBFS sets these bits to 0 when all the data is read from one FIFO buffer plane. In double buffer mode, if data is received in one FIFO buffer plane before all of the data is read from the other plane, the USBFS sets these bits to indicate the length of the receive data in the former plane when all of the data is read from the latter plane.

### FRDY bit (FIFO Port Ready)

The FRDY bit indicates whether the FIFO port can be accessed by the CPU or DMA/DTC.

In the following cases, the USBFS sets the FRDY bit to 1 but data cannot be read through the FIFO port because there is no data to be read:

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1

In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

### BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU side for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USBFS clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the DCP is the selected pipe, setting the BCLR bit to 1 allows the USBFS to clear the FIFO buffer regardless of whether the CPU or SIE has access rights. To clear the buffer when the SIE has access rights, set the DCPCTR.PID[1:0] bits to 00b (NAK response) before setting the BCLR bit to 1.

When the selected pipe is transmitting, if 1 is written to the BVAL flag and the BCLR bit simultaneously, the USBFS clears the data that is already written, enabling transmission of a zero-length packet.

When the selected pipe is not the DCP, only write 1 to the BCLR bit while the FRDY bit in the FIFO port control register is 1 (set by the USBFS).



**BVAL flag (Buffer Memory Valid Flag)**

Set the BVAL flag to 1 when data is completely written to the FIFO buffer on the CPU side for the pipe selected in CURPIPE[3:0].

When the selected pipe is transmitting, set this flag to 1 in the following cases:

- To transmit a short packet, set this flag to 1 after data is written
- To transmit a zero-length packet, set this flag to 1 before data is written to the FIFO buffer

The USBFS then switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

When data of the maximum packet size is written for the pipe in continuous transfer mode, the USBFS sets the BVAL flag to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

Only write 1 to the BVAL flag while the FRDY bit is 1 (set by the USBFS). When the selected pipe is receiving, do not set the BVAL flag to 1.

**29.2.9 DnFIFOCTR : DnFIFO Port Control Register (n = 0, 1)**

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x02A + 0x4 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BVAL	BCLR	FRDY	—	—	—	—	DTLN[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	DTLN[8:0]	Receive Data Length Indicates the receive data length. The meaning of the values differs depending on the RCNT bit setting in the port select register. For details, see the description of the DTLN[8:0] bits.	R
12:9	—	These bits are read as 0. The write value should be 0.	R/W
13	FRDY	FIFO Port Ready 0: FIFO port access disabled 1: FIFO port access enabled	R
14	BCLR	CPU Buffer Clear 0: No operation 1: Clear FIFO buffer on the CPU side	R/W <sup>1</sup>
15	BVAL	Buffer Memory Valid Flag 0: Invalid (writing 0 has no effect) 1: Writing ended	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only 0 can be read.

The CFIFOCTR, D0FIFOCTR, and D1FIFOCTR registers correspond to the CFIFO, D0FIFO, and D1FIFO buffers.

**DTLN[8:0] bits (Receive Data Length)**

The DTLN[8:0] bits indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the DnFIFOSEL.RCNT bit (n = 0, 1), as follows:

- RCNT = 0

The USBFS sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU or DMA/DTC has read all of the received data from a single FIFO buffer plane.

While the PIPECFG.BFRE bit = 1, the USBFS retains the length of the receive data until the BCLR bit is set to 1, even after all the data is read.

- RCNT = 1

The USBFS decrements the value indicated in the DTLN[8:0] bits each time data is read from the FIFO buffer. The value is decremented by 1 when MBW = 0, and by 2 when MBW = 1.

The USBFS sets these bits to 0 when all the data is read from one FIFO buffer plane. In double buffer mode, if data is received in one FIFO buffer plane before all of the data is read from the other plane, the USBFS sets these bits to indicate the length of the receive data in the former plane when all of the data is read from the latter plane.

### FRDY bit (FIFO Port Ready)

The FRDY bit indicates whether the FIFO port can be accessed by the CPU or DMA/DTC.

In the following cases, the USBFS sets the FRDY bit to 1 but data cannot be read through the FIFO port because there is no data to be read:

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty
- A short packet is received and the data is completely read while the PIPECFG.BFRE bit = 1

In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

### BCLR bit (CPU Buffer Clear)

Set the BCLR bit to 1 to clear the FIFO buffer on the CPU side for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USBFS clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the DCP is the selected pipe, setting the BCLR bit to 1 allows the USBFS to clear the FIFO buffer regardless of whether the CPU or SIE has access rights. To clear the buffer when the SIE has access rights, set the DCPCTR.PID[1:0] bits to 00b (NAK response) before setting the BCLR bit to 1.

When the selected pipe is transmitting, if 1 is written to the BVAL flag and the BCLR bit simultaneously, the USBFS clears the data that is already written, enabling transmission of a zero-length packet.

When the selected pipe is not the DCP, only write 1 to the BCLR bit while the FRDY bit in the FIFO port control register is 1 (set by the USBFS).

### BVAL flag (Buffer Memory Valid Flag)

Set the BVAL flag to 1 when data is completely written to the FIFO buffer on the CPU side for the pipe selected in CURPIPE[3:0].

When the selected pipe is transmitting, set this flag to 1 in the following cases:

- To transmit a short packet, set this flag to 1 after data is written
- To transmit a zero-length packet, set this flag to 1 before data is written to the FIFO buffer

The USBFS then switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

When data of the maximum packet size is written for the pipe in continuous transfer mode, the USBFS sets the BVAL flag to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

Only write 1 to the BVAL flag while the FRDY bit is 1 (set by the USBFS). When the selected pipe is receiving, do not set the BVAL flag to 1.

## 29.2.10 INTENB0 : Interrupt Enable Register 0

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x030

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	BRDYE	Buffer Ready Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
9	NRDYE	Buffer Not Ready Response Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
10	BEMPE	Buffer Empty Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
11	CTRE	Control Transfer Stage Transition Interrupt Enable *1 0: Disable interrupt request 1: Enable interrupt request	R/W
12	DVSE	Device State Transition Interrupt Enable *1 0: Disable interrupt request 1: Enable interrupt request	R/W
13	SOFE	Frame Number Update Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
14	RSME	Resume Interrupt Enable*1 0: Disable interrupt request 1: Enable interrupt request	R/W
15	VBSE	VBUS Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. The RSME, DVSE, and CTRE bits can only be set to 1 in device controller mode. Do not set these bits to 1 in host controller mode.

When a status flag in the INTSTS0 register sets to 1 and the associated interrupt request enable bit setting in the INTENB0 register is 1, the USBFS issues a USBFS interrupt request.

Regardless of the INTENB0 register setting, the status flag in the INTSTS0 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB0 register is switched from 0 to 1 while the associated status flag in the INTSTS0 register is set to 1, a USBFS interrupt is requested.

### 29.2.11 INTENB1 : Interrupt Enable Register 1

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x032

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	OVRC RE	BCHG E	—	DTCH E	ATTC HE	—	—	—	—	EOFE RRE	SIGNE	SACK E	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	SACK E	Setup Transaction Normal Response Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
5	SIGNE	Setup Transaction Error Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Bit	Symbol	Function	R/W
6	EOFERRE	EOF Error Detection Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
10:7	—	These bits are read as 0. The write value should be 0.	R/W
11	ATTCHE	Connection Detection Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
12	DTCHE	Disconnection Detection Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	BCHGE	USB Bus Change Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W
15	OVRCRE	Overcurrent Input Change Interrupt Enable 0: Disable interrupt request 1: Enable interrupt request	R/W

Note: S-TYPE-3, P-TYPE-3

Note: The bits in INTENB1 can only be set to 1 in host controller mode. Do not set these bits to 1 in device controller mode.

INTENB1 specifies the interrupt masks in host controller mode and for the setup transaction.

When a status flag in the INTSTS1 register sets to 1 and the associated interrupt request enable bit setting in the INTENB1 register is 1, the USBFS issues a USBFS interrupt request.

Regardless of the INTENB1 register setting, the status flag in the INTSTS1 register sets to 1 in response to a state change that satisfies the associated condition.

When an interrupt request enable bit in the INTENB1 register is switched from 0 to 1 while the associated status flag in the INTSTS1 register is set to 1, a USBFS interrupt is requested.

Do not enable interrupts in device controller mode.

### 29.2.12 BRDYENB : BRDY Interrupt Enable Register

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x036

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9 BRDY E	PIPE8 BRDY E	PIPE7 BRDY E	PIPE6 BRDY E	PIPE5 BRDY E	PIPE4 BRDY E	PIPE3 BRDY E	PIPE2 BRDY E	PIPE1 BRDY E	PIPE0 BRDY E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0BRDYE	BRDY Interrupt Enable for Pipe 0 0: Disable interrupt request 1: Enable interrupt request	R/W
1	PIPE1BRDYE	BRDY Interrupt Enable for Pipe 1 0: Disable interrupt request 1: Enable interrupt request	R/W
2	PIPE2BRDYE	BRDY Interrupt Enable for Pipe 2 0: Disable interrupt request 1: Enable interrupt request	R/W
3	PIPE3BRDYE	BRDY Interrupt Enable for Pipe 3 0: Disable interrupt request 1: Enable interrupt request	R/W

Bit	Symbol	Function	R/W
4	PIPE4BRDYE	BRDY Interrupt Enable for Pipe 4 0: Disable interrupt request 1: Enable interrupt request	R/W
5	PIPE5BRDYE	BRDY Interrupt Enable for Pipe 5 0: Disable interrupt request 1: Enable interrupt request	R/W
6	PIPE6BRDYE	BRDY Interrupt Enable for Pipe 6 0: Disable interrupt request 1: Enable interrupt request	R/W
7	PIPE7BRDYE	BRDY Interrupt Enable for Pipe 7 0: Disable interrupt request 1: Enable interrupt request	R/W
8	PIPE8BRDYE	BRDY Interrupt Enable for Pipe 8 0: Disable interrupt request 1: Enable interrupt request	R/W
9	PIPE9BRDYE	BRDY Interrupt Enable for Pipe 9 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The BRDYENB register enables or disables the INTSTS0.BRDY bit to be set to 1 when a BRDY interrupt is detected for each pipe.

When a status flag in the BRDYSTS register sets to 1 and the associated PIPE<sub>n</sub>BRDYE bit (n = 0 to 9) setting in the BRDYENB register is 1, the INTSTS0.BRDY flag sets to 1. In this case, if the BRDYE bit in INTENB0 is 1, the USBFS generates a BRDY interrupt request. While at least one PIPE<sub>n</sub>BRDY bit indicates 1, the USB generates the BRDY interrupt request when the associated interrupt request enable bit in the BRDYENB register is changed from 0 to 1 by software.

### 29.2.13 NRDYENB : NRDY Interrupt Enable Register

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x038

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	—	PIPE9 NRDY E	PIPE8 NRDY E	PIPE7 NRDY E	PIPE6 NRDY E	PIPE5 NRDY E	PIPE4 NRDY E	PIPE3 NRDY E	PIPE2 NRDY E	PIPE1 NRDY E	PIPE0 NRDY E
------------	---	---	---	---	---	---	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	PIPE0NRDYE	NRDY Interrupt Enable for Pipe 0 0: Disable interrupt request 1: Enable interrupt request	R/W
1	PIPE1NRDYE	NRDY Interrupt Enable for Pipe 1 0: Disable interrupt request 1: Enable interrupt request	R/W
2	PIPE2NRDYE	NRDY Interrupt Enable for Pipe 2 0: Disable interrupt request 1: Enable interrupt request	R/W
3	PIPE3NRDYE	NRDY Interrupt Enable for Pipe 3 0: Disable interrupt request 1: Enable interrupt request	R/W
4	PIPE4NRDYE	NRDY Interrupt Enable for Pipe 4 0: Disable interrupt request 1: Enable interrupt request	R/W

Bit	Symbol	Function	R/W
5	PIPE5NRDYE	NRDY Interrupt Enable for Pipe 5 0: Disable interrupt request 1: Enable interrupt request	R/W
6	PIPE6NRDYE	NRDY Interrupt Enable for Pipe 6 0: Disable interrupt request 1: Enable interrupt request	R/W
7	PIPE7NRDYE	NRDY Interrupt Enable for Pipe 7 0: Disable interrupt request 1: Enable interrupt request	R/W
8	PIPE8NRDYE	NRDY Interrupt Enable for Pipe 8 0: Disable interrupt request 1: Enable interrupt request	R/W
9	PIPE9NRDYE	NRDY Interrupt Enable for Pipe 9 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The NRDYENB register enables or disables the INTSTS0.NRDY bit to be set to 1 when a NRDY interrupt is detected for each pipe.

When a status flag in the NRDYSTS register sets to 1 and the associated PIPE $n$ NRDYE ( $n = 0$  to 9) bit setting in the NRDYENB register is 1, the INTSTS0.NRDY flag sets to 1. In this case, if the NRDYE bit in INTENB0 is 1, the USBFS generates a NRDY interrupt request. While at least one PIPE $n$ NRDYE bit indicates 1, the USBFS generates the NRDY interrupt request when the associated interrupt request enable bit in the NRDYENB register is changed from 0 to 1 by software.

### 29.2.14 BEMPENB : BEMP Interrupt Enable Register

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x03A

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	—	PIPE9 BEMP E	PIPE8 BEMP E	PIPE7 BEMP E	PIPE6 BEMP E	PIPE5 BEMP E	PIPE4 BEMP E	PIPE3 BEMP E	PIPE2 BEMP E	PIPE1 BEMP E	PIPE0 BEMP E
------------	---	---	---	---	---	---	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------	--------------------

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	PIPE0BEMPE	BEMP Interrupt Enable for Pipe 0 0: Disable interrupt request 1: Enable interrupt request	R/W
1	PIPE1BEMPE	BEMP Interrupt Enable for Pipe 1 0: Disable interrupt request 1: Enable interrupt request	R/W
2	PIPE2BEMPE	BEMP Interrupt Enable for Pipe 2 0: Disable interrupt request 1: Enable interrupt request	R/W
3	PIPE3BEMPE	BEMP Interrupt Enable for Pipe 3 0: Disable interrupt request 1: Enable interrupt request	R/W
4	PIPE4BEMPE	BEMP Interrupt Enable for Pipe 4 0: Disable interrupt request 1: Enable interrupt request	R/W

Bit	Symbol	Function	R/W
5	PIPE5BEMPE	BEMP Interrupt Enable for Pipe 5 0: Disable interrupt request 1: Enable interrupt request	R/W
6	PIPE6BEMPE	BEMP Interrupt Enable for Pipe 6 0: Disable interrupt request 1: Enable interrupt request	R/W
7	PIPE7BEMPE	BEMP Interrupt Enable for Pipe 7 0: Disable interrupt request 1: Enable interrupt request	R/W
8	PIPE8BEMPE	BEMP Interrupt Enable for Pipe 8 0: Disable interrupt request 1: Enable interrupt request	R/W
9	PIPE9BEMPE	BEMP Interrupt Enable for Pipe 9 0: Disable interrupt request 1: Enable interrupt request	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The BEMPENB register enables or disables the INTSTS0.BEMP bit to be set to 1 when a BEMP interrupt is detected for each pipe.

When a status flag in the BEMPSTS register sets to 1 and the associated PIPE<sub>n</sub>BEMPE (n = 0 to 9) bit setting in the BEMPENB register is 1, the INTSTS0.BEMP flag sets to 1. In this case, if the BEMPE bit in INTENB0 is 1, the USBFS generates a BEMP interrupt request. While at least one PIPE<sub>n</sub>BEMPE bit indicates 1, the USBFS generates the BEMP interrupt request when the associated interrupt request enable bit in the BEMPENB register is changed from 0 to 1 by software.

### 29.2.15 SOFCFG : SOF Output Configuration Register

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x03C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TRNE NSEL	—	BRDY M	—	EDGE STS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	EDGESTS	Edge Interrupt Output Status Monitor* <sup>1</sup> Indicates 1 during the edge processing of an edge interrupt output signal.	R
5	—	This bit is read as 0. The write value should be 0.	R/W
6	BRDYM	BRDY Interrupt Status Clear Timing 0: Clear BRDY flag by software 1: Clear BRDY flag by the USBFS through a data read from the FIFO buffer or data write to the FIFO buffer	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	TRNENSEL	Transaction-Enabled Time Select* <sup>1</sup> 0: Not low-speed communication 1: Low-speed communication	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Confirm that the EDGESTS flag is 0 before stopping the clock supply to the USBFS.

**EDGESTS bit (Edge Interrupt Output Status Monitor)**

The EDGESTS bit indicates 1 during the edge processing of an edge interrupt output signal. Confirm that this bit is 0 before stopping the clock supply to the USBFS.

**BRDYM bit (BRDY Interrupt Status Clear Timing)**

The BRDYM bit specifies how the BRDY interrupt status flags for the pipes are cleared.

**TRNENSEL bit (Transaction-Enabled Time Select)**

When the USB port is in use for full- or low-speed communications, the TRNENSEL bit specifies the timing with which the USBFS issues tokens in a frame (transaction-enabled time).

Set this bit to 1 when a low-speed device is connected. The bit is only valid in host controller mode. Set this bit to 0 in device controller mode.

**29.2.16 INTSTS0 : Interrupt Status Register 0**

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x040

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBST S	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	x	0	0	0	0	x	0	0	x	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CTSQ[2:0]	Control Transfer Stage 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error	R
3	VALID	USB Request Reception 0: Setup packet not received 1: Setup packet received	R/W
6:4	DVSQ[2:0]	Device State Indicates the device state. 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state Others: Suspend state	R
7	VBSTS	VBUS Input Status 0: USB_VBUS pin is low 1: USB_VBUS pin is high	R
8	BRDY	Buffer Ready Interrupt Status 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R
9	NRDY	Buffer Not Ready Interrupt Status 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R
10	BEMP	Buffer Empty Interrupt Status 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R
11	CTRT	Control Transfer Stage Transition Interrupt Status*2 0: No control transfer stage transition interrupt occurred 1: Control transfer stage transition interrupt occurred	R/W*1



Bit	Symbol	Function	R/W
12	DVST	Device State Transition Interrupt Status <sup>*2</sup> 0: No device state transition interrupt occurred 1: Device state transition interrupt occurred	R/W <sup>*1</sup>
13	SOFR	Frame Number Refresh Interrupt Status 0: No SOF interrupt occurred 1: SOF interrupt occurred	R/W <sup>*1</sup>
14	RESM	Resume Interrupt Status <sup>*2 *3</sup> 0: No resume interrupt occurred 1: Resume interrupt occurred	R/W <sup>*1</sup>
15	VBINT	VBUS Interrupt Status <sup>*3</sup> 0: No VBUS interrupt occurred 1: VBUS interrupt occurred	R/W <sup>*1</sup>

Note: S-TYPE-3, P-TYPE-3

Note: The value of the DVST bit is 0 when the MCU is reset and 1 after a USB bus reset.

Note: The value of the VBSTS bit is 1 when the USB\_VBUS pin is high and 0 when the USB\_VBUS pin is low.

Note: The value of the DVSQ[2:0] bits is 000b when the MCU is reset and 001b after a USB bus reset.

Note 1. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bits, write 0 only to the bits to be cleared. Write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Note 2. The status of the RESM, DVST, and CTRT bits are changed only in device controller mode. Set the associated interrupt enable bits to 0 (disabled) in host controller mode.

Note 3. The USBFS detects a change in the status indicated in the VBINT and RESM bits even while the clock supply is stopped (SYSCFG.SCKE bit = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply before clearing the status by software.

### CTSQ[2:0] bits (Control Transfer Stage)

In host controller mode, the read value of the CTSQ[2:0] bits is invalid.

### VALID bit (USB Request Reception)

In host controller mode, the read value of the VALID bit is invalid.

### DVSQ[2:0] bits (Device State)

The DVSQ[2:0] bits are initialized by a USB bus reset. In host controller mode, the read value is invalid.

### BRDY flag (Buffer Ready Interrupt Status)

The BRDY flag indicates the BRDY interrupt status.

The USBFS sets the BRDY bit to 1 when it detects a BRDY interrupt status (PIPE<sub>n</sub>BRDY = 1, n = 0 to 9) on at least one pipe for which BRDY interrupts are enabled (BRDYENB.PIPE<sub>n</sub>BRDYE = 1).

For the conditions that cause the PIPE<sub>n</sub>BRDY status to be asserted, see [section 29.3.3.1. BRDY interrupt](#).

The USBFS sets the BRDY bit to 0 when the software writes 0 to all of the PIPE<sub>n</sub>BRDY bits associated with the PIPE<sub>n</sub>BRDYE bits that are set to 1. Writing 0 to the BRDY flag in the software does not clear the flag.

### NRDY flag (Buffer Not Ready Interrupt Status)

The NRDY flag indicates the NRDY interrupt status.

The USBFS sets the NRDY bit to 1 when it detects a NRDY interrupt status (PIPE<sub>n</sub>NRDY = 1, n = 0 to 9) on at least one pipe for which NRDY interrupts are enabled (NRDYENB.PIPE<sub>n</sub>NRDYE = 1).

For the conditions that cause the PIPE<sub>n</sub>NRDY status to be asserted, see [section 29.3.3.2. NRDY interrupt](#).

The USBFS sets the NRDY bit to 0 when the software writes 0 to all of the PIPE<sub>n</sub>NRDY bits associated with the PIPE<sub>n</sub>NRDYE bits that are set to 1. Writing 0 to the NRDY flag in the software does not clear the flag.

### BEMP flag (Buffer Empty Interrupt Status)

The BEMP flag indicates the BEMP interrupt status.

The USBFS sets the BEMP bit to 1 when it detects a BEMP interrupt status (PIPE<sub>n</sub>BEMP = 1, n = 0 to 9) on at least one pipe for which BEMP interrupts are enabled (BEMPENB.PIPE<sub>n</sub>BEMPE = 1).

For the conditions that cause the PIPE<sub>n</sub>BEMP status to be asserted, see [section 29.3.3.3. BEMP interrupt](#).

The USBFS sets the BEMP bit to 0 when the software writes 0 to all of the PIPE<sub>n</sub>BEMP bits associated with the PIPE<sub>n</sub>BEMPE bits that are set to 1. Writing 0 to the BEMP flag in the software does not clear the flag.

**CTRT flag (Control Transfer Stage Transition Interrupt Status)**

In device controller mode, the USBFS updates the value of the CTSQ[2:0] bits and sets the CTRT flag to 1 on detecting a transition in the control transfer stage. When a control transfer stage transition interrupt occurs, clear the CTRT flag before the USBFS detects the next control transfer stage transition.

Values read from the CTRT flag in host controller mode are invalid.

**DVST flag (Device State Transition Interrupt Status)**

In device controller mode, the USBFS updates the value of the DVSQ[2:0] bits and sets the DVST flag to 1 on detecting a change in the device state. When a device state transition interrupt occurs, clear the DVST flag before the USBFS detects the next device state transition.

Values read from the DVST flag in host controller mode are invalid.

**SOFR flag (Frame Number Refresh Interrupt Status)**

In host controller mode, the USBFS sets the SOFR flag to 1 on updating the frame number when the DVSTCTR0.UACT bit is set to 1 by software. A SOFR interrupt is detected every 1 ms.

In device controller mode, the USBFS sets the SOFR flag to 1 on updating the frame number. A frame number refresh interrupt is detected every 1 ms.

The USBFS can detect an SOFR interrupt through the internal interpolation function even when a corrupted SOF packet is received from the USB host.

**RESM flag (Resume Interrupt Status)**

In device controller mode, the USBFS sets the RESM flag to 1 on detecting the falling edge of the signal on the USB\_DP pin in the Suspend state (DVSQ[2:0] = 1xxb). Values read from the RESM flag in host controller mode are invalid.

**VBINT flag (VBUS Interrupt Status)**

The USBFS sets the VBINT flag to 1 on detecting a level change (high to low or low to high) in the USB\_VBUS pin input value. The USBFS sets the VBSTS flag to indicate the USB\_VBUS pin input value. When a VBUS interrupt occurs, eliminate transient elements by reading the VBSTS flag at least three times through software processing and check that the values read are the same.

**29.2.17 INTSTS1 : Interrupt Status Register 1**

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x042

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRC R	BCHG	—	DTCH	ATTC H	—	—	—	—	EOFE RR	SIGN	SACK	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	SACK	Setup Transaction Normal Response Interrupt Status 0: No SACK interrupt occurred 1: SACK interrupt occurred	R/W <sup>1</sup>
5	SIGN	Setup Transaction Error Interrupt Status 0: No SIGN interrupt occurred 1: SIGN interrupt occurred	R/W <sup>1</sup>
6	EOFERR	EOF Error Detection Interrupt Status 0: No EOFERR interrupt occurred 1: EOFERR interrupt occurred	R/W <sup>1</sup>

Bit	Symbol	Function	R/W
10:7	—	These bits are read as 0. The write value should be 0.	R/W
11	ATTCH	ATTCH Interrupt Status 0: No ATTCH interrupt occurred 1: ATTCH interrupt occurred	R/W <sup>*1</sup>
12	DTCH	USB Disconnection Detection Interrupt Status 0: No DTCH interrupt occurred 1: DTCH interrupt occurred	R/W <sup>*1</sup>
13	—	This bit is read as 0. The write value should be 0.	R/W
14	BCHG	USB Bus Change Interrupt Status <sup>*2</sup> 0: No BCHG interrupt occurred 1: BCHG interrupt occurred	R/W <sup>*1</sup>
15	OVRRCR	Overcurrent Input Change Interrupt Status <sup>*2</sup> 0: No OVRRCR interrupt occurred 1: OVRRCR interrupt occurred	R/W <sup>*1</sup>

Note: S-TYPE-3, P-TYPE-3

Note 1. To clear the bits in INTSTS1, write 0 only to the bits to be cleared. Write 1 to the other bits.

Note 2. The USBFS detects a change in the status in the OVRRCR or BCHG bit even when the clock supply is stopped (SYSCFG.SCKE = 0), and it requests the interrupt when the associated interrupt request bit is 1. Enable the clock supply (SYSCFG.SCKE = 1) before clearing the status through the software. No other interrupts can be detected while the clock supply is stopped (SYSCFG.SCKE bit = 0).

INTSTS1 is used to confirm the status of each interrupt in host controller mode. Only enable the status change interrupts indicated in the bits in INTSTS1 in host controller mode.

### SACK flag (Setup Transaction Normal Response Interrupt Status)

The SACK flag indicates the status of the setup transaction normal response interrupt in host controller mode.

The USBFS detects the SACK interrupt and sets this flag to 1 when an ACK response is returned from a peripheral device during the setup transactions issued by the USBFS. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

Values read from the SACK flag in device controller mode are invalid.

### SIGN flag (Setup Transaction Error Interrupt Status)

The SIGN flag indicates the status of setup transaction error interrupts in host controller mode.

The USBFS detects the SIGN interrupt and sets this flag to 1 when an ACK response is not returned from a peripheral device three consecutive times during the setup transactions issued by the USBFS. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects the SIGN interrupt when any of the following response conditions occur for three consecutive setup transactions:

- Timeout is detected by the USBFS when the peripheral device has returned no response
- A corrupted ACK packet is received
- A handshake other than ACK (NAK, NYET, or STALL) is received

Values read from the SIGN flag in device controller mode are invalid.

### EOFERR flag (EOF Error Detection Interrupt Status)

The EOFERR flag indicates the status of EOF error detection interrupts in host controller mode.

The USBFS detects the EOFERR interrupt and sets this flag to 1 on detecting that communication did not complete at the EOF2 timing defined in the USB 2.0 specification. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

After detecting the EOFERR interrupt, the USBFS controls the hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt was detected to 0
- Puts the port in which the EOFERR interrupt occurred into the idle state

The software must terminate all pipes in which communications are being carried out and re-enumerate the USB port. Values read from the EOFERR flag in device controller mode are invalid.

#### **ATTCH flag (ATTCH Interrupt Status)**

The ATTCH flag indicates the status of USB attach detection interrupts in host controller mode.

The USBFS detects the ATTCH interrupt and sets this flag to 1 on detecting a J- or K-state on the full- or low-speed signal level for 2.5  $\mu$ s. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects the ATTCH interrupt on any of the following conditions.

- K-state, SE0, or SE1 changes to J-state, and J-state continues for 2.5  $\mu$ s
- J-state, SE0, or SE1 changes to K-state, and K-state continues for 2.5  $\mu$ s

Values read from the ATTCH flag in device controller mode are invalid.

#### **DTCH flag (USB Disconnection Detection Interrupt Status)**

The DTCH flag indicates the status of USB disconnection detection interrupts in host controller mode.

The USBFS detects the DTCH interrupt and sets this flag to 1 on detecting a USB bus detach event. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS detects bus detach events based on the USB 2.0 specification.

After detecting the DTCH interrupt, the USBFS controls hardware as follows, regardless of the associated interrupt enable bit setting:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt was detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state

The software must terminate all pipes in which communications are being carried out and transition to a wait state for connecting to the USB port (waiting for ATTCH interrupt generation).

Values read from the DTCH flag in device controller mode are invalid.

#### **BCHG flag (USB Bus Change Interrupt Status)**

The BCHG flag indicates the status of USB bus change interrupts in host controller mode.

The USBFS detects the BCHG interrupt and sets this flag to 1 when a change in the full- or low-speed signal level occurs on the USB port. This includes any change from J-state, K-state, or SE0 to J-state, K-state, or SE0. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

The USBFS sets the LNST[1:0] flags to indicate the input state of the USB port. When a BCHG interrupt occurs, eliminate transient elements by repeat reading the LNST[1:0] flags by software until the same value is read at least three times.

Change in the USB bus state can be detected while the internal clock is stopped.

Values read from the BCHG flag in device controller mode are invalid.

#### **OVRCCR flag (Overcurrent Input Change Interrupt Status)**

The OVRCCR flag indicates the status of USB\_OVRCURA, USB\_OVRCURA-DS, USB\_OVRCURB or USB\_OVRCURB-DS input pin change interrupts.

The USBFS detects the OVRCCR interrupt and sets this flag to 1 when a change (high to low or low to high) occurs in at least one of the input values to the USB\_OVRCURA, USB\_OVRCURA-DS, USB\_OVRCURB or USB\_OVRCURB-DS pins. If the associated interrupt enable bit is set to 1 by software, the USBFS generates the interrupt.

## 29.2.18 BRDYSTS : BRDY Interrupt Status Register

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x046

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9 BRDY	PIPE8 BRDY	PIPE7 BRDY	PIPE6 BRDY	PIPE5 BRDY	PIPE4 BRDY	PIPE3 BRDY	PIPE2 BRDY	PIPE1 BRDY	PIPE0 BRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0BRDY	BRDY Interrupt Status for Pipe 0 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
1	PIPE1BRDY	BRDY Interrupt Status for Pipe 1 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
2	PIPE2BRDY	BRDY Interrupt Status for Pipe 2 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
3	PIPE3BRDY	BRDY Interrupt Status for Pipe 3 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
4	PIPE4BRDY	BRDY Interrupt Status for Pipe 4 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
5	PIPE5BRDY	BRDY Interrupt Status for Pipe 5 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
6	PIPE6BRDY	BRDY Interrupt Status for Pipe 6 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
7	PIPE7BRDY	BRDY Interrupt Status for Pipe 7 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
8	PIPE8BRDY	BRDY Interrupt Status for Pipe 8 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
9	PIPE9BRDY	BRDY Interrupt Status for Pipe 9 *2 0: No BRDY interrupt occurred 1: BRDY interrupt occurred	R/W*1
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated in the bits in BRDYSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

Note 2. When the SOFCFG.BRDYM bit is set to 0, clear BRDY interrupts before accessing the FIFO.

### 29.2.19 NRDYSTS : NRDY Interrupt Status Register

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x048

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9 NRDY	PIPE8 NRDY	PIPE7 NRDY	PIPE6 NRDY	PIPE5 NRDY	PIPE4 NRDY	PIPE3 NRDY	PIPE2 NRDY	PIPE1 NRDY	PIPE0 NRDY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0NRDY	NRDY Interrupt Status for Pipe 0 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W <sup>1</sup>
1	PIPE1NRDY	NRDY Interrupt Status for Pipe 1 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W <sup>1</sup>
2	PIPE2NRDY	NRDY Interrupt Status for Pipe 2 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W <sup>1</sup>
3	PIPE3NRDY	NRDY Interrupt Status for Pipe 3 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W <sup>1</sup>
4	PIPE4NRDY	NRDY Interrupt Status for Pipe 4 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W <sup>1</sup>
5	PIPE5NRDY	NRDY Interrupt Status for Pipe 5 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W <sup>1</sup>
6	PIPE6NRDY	NRDY Interrupt Status for Pipe 6 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W <sup>1</sup>
7	PIPE7NRDY	NRDY Interrupt Status for Pipe 7 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W <sup>1</sup>
8	PIPE8NRDY	NRDY Interrupt Status for Pipe 8 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W <sup>1</sup>
9	PIPE9NRDY	NRDY Interrupt Status for Pipe 9 0: No NRDY interrupt occurred 1: NRDY interrupt occurred	R/W <sup>1</sup>
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. To clear the status indicated in the bits in NRDYSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

### 29.2.20 BEMPSTS : BEMP Interrupt Status Register

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x04A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPE9 BEMP	PIPE8 BEMP	PIPE7 BEMP	PIPE6 BEMP	PIPE5 BEMP	PIPE4 BEMP	PIPE3 BEMP	PIPE2 BEMP	PIPE1 BEMP	PIPE0 BEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIPE0BEMP	BEMP Interrupt Status for Pipe 0 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W <sup>1</sup>
1	PIPE1BEMP	BEMP Interrupt Status for Pipe 1 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W <sup>1</sup>
2	PIPE2BEMP	BEMP Interrupt Status for Pipe 2 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W <sup>1</sup>
3	PIPE3BEMP	BEMP Interrupt Status for Pipe 3 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W <sup>1</sup>
4	PIPE4BEMP	BEMP Interrupt Status for Pipe 4 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W <sup>1</sup>
5	PIPE5BEMP	BEMP Interrupt Status for Pipe 5 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W <sup>1</sup>
6	PIPE6BEMP	BEMP Interrupt Status for Pipe 6 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W <sup>1</sup>
7	PIPE7BEMP	BEMP Interrupt Status for Pipe 7 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W <sup>1</sup>
8	PIPE8BEMP	BEMP Interrupt Status for Pipe 8 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W <sup>1</sup>
9	PIPE9BEMP	BEMP Interrupt Status for Pipe 9 0: No BEMP interrupt occurred 1: BEMP interrupt occurred	R/W <sup>1</sup>
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. To clear the status indicated in the bits in BEMPSTS, write 0 only to the bits to be cleared. Write 1 to the other bits.

### 29.2.21 FRMNUM : Frame Number Register

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x04C

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
10:0	FRNM[10:0]	Frame Number Latest frame number.	R
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	CRCE	Receive Data Error 0: No error occurred 1: Error occurred	R/W <sup>1</sup>
15	OVRN	Overflow/Underflow Detection Status 0: No error occurred 1: Error occurred	R/W <sup>1</sup>

Note: S-TYPE-3, P-TYPE-3

Note 1. To clear the status, write 0 only to the bits to be cleared. Write 1 to the other bits.

### FRNM[10:0] flags (Frame Number)

The USBFS sets the FRNM[10:0] flags to indicate the latest frame number, which is updated every 1 ms, when an SOF packet is issued or received.

### CRCE flag (Receive Data Error)

The CRCE flag sets to 1 when a CRC error or bit stuffing error occurs during isochronous transfer. On detecting a CRC error in host controller mode, the USBFS generates an internal NRDY interrupt.

To clear the CRCE flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

### OVRN flag (Overrun/Underrun Detection Status)

The OVRN flag sets to 1 when an overrun or underrun error occurs during isochronous transfer. To clear the flag, write 0 to it while writing 1 to the other bits in the FRMNUM register.

In host controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the time to issue an OUT token comes before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the time to issue an IN token comes when no FIFO buffer planes are empty

In device controller mode, the OVRN flag sets to 1 on any of the following conditions:

- For a transmitting isochronous pipe, the IN token is received before all of the transmit data is written to the FIFO buffer
- For a receiving isochronous pipe, the OUT token is received when no FIFO buffer planes are empty

## 29.2.22 DVCHGR : Device State Change Register

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x04E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DVCHG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0. The write value should be 0.	R/W
15	DVCHG	Device State Change 0: Disable writes to the USBADDR.STSRECOV[3:0] and USBADDR.USBADDR[6:0] bits 1: Enable writes to the USBADDR.STSRECOV[3:0] and USBADDR.USBADDR[6:0] bits	R/W

Note: S-TYPE-3, P-TYPE-3

For details, see [section 29.3.1.5. Release from Deep Software Standby mode1 because of USB suspend/resume interrupts.](#)

## 29.2.23 USBADDR : USB Address Register

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x050

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	STSRECOV[3:0]			—	USBADDR[6:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Bit	Symbol	Function	R/W
6:0	USBADDR[6:0]	USB Address In device controller mode, these bits indicate the USB address assigned by the host when the USBFS processed the SET_ADDRESS request successfully.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
11:8	STSRECOV[3:0]	Status Recovery  0x4: Recovery in device controller mode: Setting prohibited Recovery in host controller mode: Return to the low-speed state (bits DVSTCTR0.RHST[2:0] = 001b) 0x8: Recovery in device controller mode: Setting prohibited Recovery in host controller mode: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b) 0x9: Recovery in device controller mode: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 001b (default state) Recovery in host controller mode: Setting prohibited 0xA: Recovery in device controller mode: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 010b (address state) Recovery in host controller mode: Setting prohibited 0xB: Recovery in device controller mode: Return to the full-speed state (bits DVSTCTR0.RHST[2:0] = 010b), bits INTSTS0.DVSQ[2:0] = 011b (configured state) Recovery in host controller mode: Setting prohibited Others: Setting prohibited	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

### USBADDR[6:0] bits (USB Address)

In device controller mode, the USBADDR[6:0] flags indicate the USB address received when the USBFS processed a SetAddress request successfully. The USBFS sets the USBADDR[6:0] bits to 0x00 on detecting a USB bus reset.

Writing to these bits is enabled while the DVCHGR.DVCHG bit is set to 1. On recovering from a USB power shut-off, the operation can resume from the USB address set before the software shut-off.

In host controller mode, the USBADDR[6:0] bits are invalid.

### STSRECOV[3:0] bits (Status Recovery)

Use the STSRECOV[3:0] bits to resume the state of the internal sequencer on recovering from USB power shut-off. For details, see [section 29.3.1.5. Release from Deep Software Standby mode1 because of USB suspend/resume interrupts.](#)

Writing to these bits is enabled while the DVCHGR.DVCHG bit is set to 1.

## 29.2.24 USBREQ : USB Request Type Register

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x054

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	BMREQUESTTYPE[7:0]	Request Type USB request bmRequestType value	R/W <sup>1</sup>
15:8	BREQUEST[7:0]	Request USB request bRequest value	R/W <sup>1</sup>

Note: S-TYPE-3, P-TYPE-3

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

USBREQ stores setup requests for control transfers.

In device controller mode, the USBREQ stores the received bRequest and bmRequestType values. In host controller mode, it sets to the bRequest and bmRequestType values to be transmitted.

USBREQ is initialized by a USB bus reset.

**BMREQUESTTYPE[7:0] bits (Request Type)**

The BMREQUESTTYPE[7:0] bits hold the bmRequestType value of USB requests.

- In host controller mode:  
Set these bits to the value of the USB request data in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:  
These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

**BREQUEST[7:0] bits (Request)**

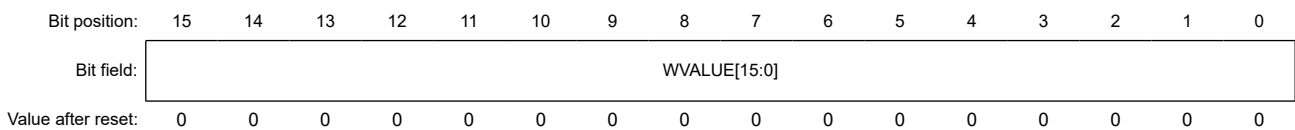
The BREQUEST[7:0] bits store bRequest value of the USB request.

- In host controller mode:  
Set these bits to the value of the USB request data in setup transmission transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:  
These bits indicate the value of the USB request data in reception setup transactions. Writing to the bits has no effect.

**29.2.25 USBVAL : USB Request Value Register**

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x056



Bit	Symbol	Function	R/W
15:0	WVALUE[15:0]	Value USB request wValue value	R/W <sup>1</sup>

Note: S-TYPE-3, P-TYPE-3

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

In device controller mode, USBVAL stores the received wValue value. In host controller mode, it sets to the wValue value to be transmitted is set.

USBVAL is initialized by a USB bus reset.

**WVALUE[15:0] bits (Value)**

The WVALUE[15:0] bits store wValue value of the USB request.

- In host controller mode:  
Set these bits to the value of the wValue field in USB requests of transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:

These bits indicate the wValue value of USB requests in reception setup transactions. Writing to the bits has no effect.

### 29.2.26 USBINDX : USB Request Index Register

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x058

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WINDEX[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WINDEX[15:0]	Index USB request wIndex value	R/W <sup>1</sup>

Note: S-TYPE-3, P-TYPE-3

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

USBINDX stores setup requests for control transfers.

In device controller mode, it stores the received wIndex value. In host controller mode, it sets to the wIndex value to be transmitted.

USBINDX is initialized by a USB bus reset.

#### WINDEX[15:0] bits (Index)

The WINDEX[15:0] bits hold the wIndex value of a USB request.

- In host controller mode:
 

Set these bits to the wIndex value in USB requests in transmission setup transactions. Do not change the value of the bits while the DPCCTR.SUREQ bit is 1.
- In device controller mode:
 

These bits indicate the wIndex value in USB requests received in reception setup transactions. Writing to the bits has no effect.

### 29.2.27 USBLENG : USB Request Length Register

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x05A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WLENTUH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WLENTUH[15:0]	Length USB request wLength value	R/W <sup>1</sup>

Note: S-TYPE-3, P-TYPE-3

Note 1. In device controller mode, these bits can be read, but writing to them has no effect. In host controller mode, these bits are both read/write bits.

USBLENG stores setup requests for control transfers.

In device controller mode, the value of wLength that is received is stored. In host controller mode, the value of wLength to be transmitted is set.

USBLENG is initialized by a USB bus reset.

**WLENTUH[15:0] bits (Length)**

The WLENTUH[15:0]bits hold the wLength value of a USB request.

- In host controller mode:  
Set these bits to the wLength value in USB requests in transmission setup transactions. Do not change the value of the bits while the DCPCTR.SUREQ bit is 1.
- In device controller mode:  
These bits indicate the wLength value in USB requests received in reception setup transactions. Writing to the bits has no effect.

**29.2.28 DCPCFG : DCP Configuration Register**

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x05C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SHTN AK	—	—	DIR	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	DIR	Transfer Direction*1 0: Data receiving direction 1: Data transmitting direction	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	SHTNAK	Pipe Disabled at End of Transfer*1 0: Keep pipe open after transfer ends 1: Disable pipe after transfer ends	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set this bit while the PID is NAK. Before setting this bit, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

**DIR bit (Transfer Direction)**

In host controller mode, the DIR bit sets the transfer direction of the data stage and status stage for control transfers. In device controller mode, set the DIR bit to 0.

**SHTNAK bit (Pipe Disabled at End of Transfer)**

The SHTNAK bit specifies whether to change PID to NAK on transfer end when the selected pipe is receiving. It is only valid when the selected pipe is receiving.

When the SHTNAK bit is 1, the USBFS changes the DCPCTR.PID[1:0] bits for the DCP to NAK on determining that a transfer has ended. The USBFS determines transfer end on the following condition:

- A short packet, including a zero-length packet, is successfully received.

### 29.2.29 DCPMAXP : DCP Maximum Packet Size Register

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x05E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DEVSEL[3:0]			—	—	—	—	—	MXPS[6:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	MXPS[6:0]	Maximum Packet Size*1 Maximum data payload specification (maximum packet size) for the DCP	R/W
11:7	—	These bits are read as 0. The write value should be 0.	R/W
15:12	DEVSEL[3:0]	Device Select*2 0x0: Address 0000b 0x1: Address 0001b 0x2: Address 0010b 0x3: Address 0011b 0x4: Address 0100b 0x5: Address 0101b Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the MXPS[6:0] bits while PID is NAK. Before setting these bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary. After the MXPS[6:0] bits are set and the DCP is set to the CURPIPE[3:0] bits in a port select register, clear the buffer by setting the BCLR bit in the port control register to 1.

Note 2. Only set the DEVSEL[3:0] bits while PID is NAK and the DCPCTR.SUREQ bit is 0. Before setting these bits, check that the DCPCTR.PBUSY bit is 0, and then change the DCPCTR.PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

#### MXPS[6:0] bits (Maximum Packet Size)

The MXPS[6:0] bits specify the maximum data payload (maximum packet size) for the DCP. The initial value is 0x40 (64 bytes). Set the bits to a USB 2.0-compliant value. Do not write to the FIFO buffer or set PID = BUF while MXPS[6:0] is set to 0.

#### DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target peripheral device for a control transfer. Set up the associated DEVADDn (n = 0 to 5) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0010b, for example, first set the address in the DEVADD2 register.

In device controller mode, set these bits to 0000b.

### 29.2.30 DCPCTR : DCP Control Register

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x060

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	SUREQ	—	—	SUREQLR	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
2	CCPL	Control Transfer End Enable 0: Disable control transfer completion 1: Enable control transfer completion	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	Pipe Busy 0: DCP not used for the USB bus 1: DCP in use for the USB bus	R
6	SQMON	Sequence Toggle Bit Monitor 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set* <sup>2</sup> Sets the sequence toggle bit in DCP transfers. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1	R/W* <sup>1</sup>
8	SQCLR	Sequence Toggle Bit Clear* <sup>2</sup> Clears the sequence toggle bit in DCP transfers. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	R/W* <sup>1</sup>
10:9	—	These bits are read as 0. The write value should be 0.	R/W
11	SUREQCLR	SUREQ Bit Clear Clears the SUREQ bit in host controller mode. 0: Invalid (writing 0 has no effect) 1: Clear SUREQ to 0	R/W* <sup>1</sup>
13:12	—	These bits are read as 0. The write value should be 0.	R/W
14	SUREQ	Setup Token Transmission Sets up token transmission in host controller mode. 0: Invalid (writing 0 has no effect) 1: Transmit setup packet	R/W
15	BSTS	Buffer Status 0: Buffer access disabled 1: Buffer access enabled	R

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is read as 0.

Note 2. Only set the SQSET and SQCLR bits while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits for the DCP from BUF to NAK. If the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

### PID[1:0] bits (Response PID)

The PID[1:0] bits control the USB response type during control transfers.

In host controller mode, to change the PID[1:0] setting from NAK to BUF:

- When the transmitting direction is set:
  - a. Write all of the transmit data to the FIFO buffer while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  - b. Set PID[1:0] bits to 01b (BUF).  
The USBFS then executes the OUT transaction.
- When the receiving direction is set:
  - a. Check that the FIFO buffer is empty (or empty the buffer) while the DVSTCTR0.UACT bit is 1 and PID is NAK.
  - b. Set PID[1:0] bits to 01b (BUF).  
The USBFS then executes the IN transaction.

The USBFS changes the PID[1:0] setting as follows:

- When the PID[1:0] bits are set to BUF (01b) by software and the USBFS has received data exceeding MaxPacketSize, the USBFS sets the PID[1:0] to STALL (11b)
- When a reception error, such as a CRC error, is detected three times consecutively, the USBFS sets the PID[1:0] bits to NAK (00b)
- On receiving the STALL handshake, the USBFS sets PID[1:0] to STALL (11b)

In device controller mode, the USBFS changes the PID[1:0] setting as follows:

- On receiving a setup packet, the USBFS sets PID[1:0] to NAK (00b). The USBFS then sets the INTSTS0.VALID flag to 1, and the PID[1:0] setting cannot be changed until the software clears the VALID flag to 0.
- When the PID[1:0] bits are set to BUF (01b) by software and the USBFS has received data exceeding MaxPacketSize, the USBFS sets PID[1:0] to STALL (11b)
- On detecting a control transfer sequence error, the USBFS sets PID[1:0] to STALL (1xb)
- On detecting a USB bus reset, the USBFS sets PID[1:0] to NAK

The USBFS does not check the PID[1:0] setting while processing a SET\_ADDRESS request.

The PID[1:0] bits are initialized by a USB bus reset.

### CCPL bit (Control Transfer End Enable)

In device controller mode, setting the CCPL bit to 1 enables the status stage of the control transfer to be completed. When the bit is set to 1 by software while the associated PID[1:0] bits are set to BUF, the USBFS completes the control transfer status stage.

During control read transfers, the USBFS transmits the ACK handshake in response to the OUT transaction from the USB host. During control write or no-data control transfers, it transmits the zero-length packet in response to the IN transaction from the USB host. On detecting a SET\_ADDRESS request, the USBFS operates in auto response mode from the setup stage up to status stage completion regardless of the CCPL bit setting.

The USBFS changes the CCPL bit from 1 to 0 on receiving a new setup packet. The software cannot write 1 to the bit while the INTSTS0.VALID bit is 1. The bit is initialized by a USB bus reset.

In host controller mode, always write 0 to the CCPL bit.

### PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether DCP is used for the transaction when USBFS changes the PID[1:0] bits from BUF to NAK. The USBFS changes the PBUSY bit from 0 to 1 on start of a USB transaction for the selected pipe. It changes the PBUSY bit from 1 to 0 on completion of one transaction.

After PID is set to NAK by software, the value in the PBUSY bit indicates whether changes to pipe settings can proceed.

For details, see [section 29.3.4.1. Pipe control register switching procedures](#).

### SQMON bit (Sequence Toggle Bit Monitor)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

The USBFS toggles the bit on normal completion of the transaction. It does not toggle the bit, however, when a DATAPID mismatch occurs during a transfer in the receiving direction.

In device controller mode, the USBFS sets the SQMON bit to 1 (specifies DATA1 as the expected value) on successful reception of the setup packet.

In device controller mode, the USBFS does not reference this bit during IN or OUT transactions at the status stage, and it does not toggle the bit on normal completion.

### SQSET bit (Sequence Toggle Bit Set)

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

**SQCLR bit (Sequence Toggle Bit Clear)**

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during a DCP transfer. It is read as 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

**SUREQCLR bit (SUREQ Bit Clear)**

In host controller mode, setting the SUREQCLR bit to 1 clears the SUREQ bit to 0. The bit is read as 0.

If transfer stops while the SUREQ bit is set to 1 in a setup transaction, set the SUREQCLR bit to 1 by software. This is not necessary at the end of a normal setup transaction, because the USBFS automatically clears the SUREQ bit to 0.

Only control the SUREQ bit through the SUREQCLR bit while the DVSTCTR0.UACT bit is 0. When UACT is 0, communication is halted or no transfer is occurring because a bus disconnection was detected.

In device controller mode, always write 0 to this bit.

**SUREQ bit (Setup Token Transmission)**

In host controller mode, setting the SUREQ bit to 1 triggers the USBFS to transmit the setup packet. After completing the setup transaction process, the USBFS generates either the SACK or SIGN interrupt and clears the SUREQ bit to 0. The USBFS also clears the SUREQ bit to 0 when the software sets the SUREQCLR bit to 1.

Before setting the SUREQ bit to 1, set the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, and USBLENG appropriately to transmit the target USB request in the setup transaction. Also check that the PID[1:0] bits for the DCP are set to NAK. After setting the SUREQ bit to 1, do not change the DCPMAXP.DEVSEL[3:0] bits, USBREQ, USBVAL, USBINDX, or USBLENG until the setup transaction is complete (SUREQ bit = 1). Write 1 to the SUREQ bit only when transmitting the setup token. Otherwise, write 0.

In device controller mode, always write 0 to this bit.

**BSTS flag (Buffer Status)**

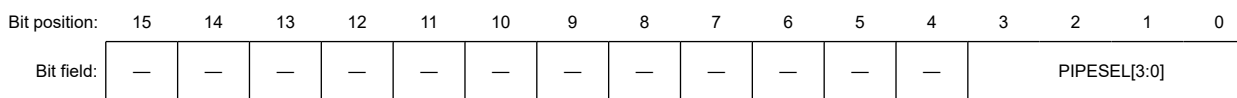
The BSTS flag indicates the status of access to the DCP FIFO buffer. The meaning of this flag varies as follows depending on the CFIFOSEL.ISEL setting:

- When ISEL = 0, the bit indicates whether receive data can be read from the buffer
- When ISEL = 1, the bit indicates whether transmit data can be written to the buffer

**29.2.31 PIPESEL : Pipe Window Select Register**

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x064



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
3:0	PIPESEL[3:0]	Pipe Window Select 0x0: No pipe selected 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W



Note: S-TYPE-3, P-TYPE-3

Set pipes 1 to 9 using the PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN registers (n = 1 to 9).

After selecting the pipe in the PIPESEL register, pipe functions must be set in the associated PIPECFG, PIPEMAXP, and PIPEPERI registers. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set independently of the pipe selection in this register.

### PIPESEL[3:0] bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number associated with the PIPECFG, PIPEMAXP, and PIPEPERI registers used for data writing and reading. Selecting a pipe number in the PIPESEL[3:0] bits allows writing to and reading from PIPECFG, PIPEMAXP, and PIPEPERI associated with the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in PIPECFG, PIPEMAXP, and PIPEPERI. Writing to these bits has no effect.

## 29.2.32 PIPECFG : Pipe Configuration Register

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x068

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TYPE[1:0]		—	—	—	BFRE	DBLB	—	SHTN AK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	EPNUM[3:0]	Endpoint Number* <sup>1</sup> Specifies the endpoint number for the selected pipe. Setting 0000b indicates that the pipe is not used.	R/W
4	DIR	Transfer Direction* <sup>2</sup> * <sup>3</sup> 0: Receiving direction 1: Transmitting direction	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	SHTNAK	Pipe Disabled at End of Transfer* <sup>1</sup> 0: Continue pipe operation after transfer ends 1: Disable pipe after transfer ends	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	DBLB	Double Buffer Mode* <sup>2</sup> * <sup>3</sup> 0: Single buffer 1: Double buffer	R/W
10	BFRE	BRDY Interrupt Operation Specification* <sup>2</sup> * <sup>3</sup> 0: Generate BRDY interrupt on transmitting or receiving data 1: Generate BRDY interrupt on completion of reading data	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W
15:14	TYPE[1:0]	Transfer Type* <sup>1</sup> 0 0: Pipe not used 0 1: Pipes 1 and 2: Bulk transfer Pipes 3 to 5: Bulk transfer Pipes 6 to 9: Setting prohibited 1 0: Pipes 1 and 2: Setting prohibited Pipes 3 to 5: Setting prohibited Pipes 6 to 9: Interrupt transfer 1 1: Pipes 1 and 2: Isochronous transfer Pipes 3 to 5: Setting prohibited Pipes 6 to 9: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

- Note 1. Only set the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.
- Note 2. Only set the BFRE, DBLB, and DIR bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.
- Note 3. To change the BFRE, DBLB, or DIR bits after completing USB communication on the selected pipe, in addition to the constraints described in Note 2, write 1 and 0 to the PIPEnCTR.ACLRM bit continuously through the software and clear the FIFO buffer assigned to the pipe.

PIPECFG specifies the transfer type, FIFO buffer access direction, and endpoint numbers for pipes 1 to 9. It also selects single or double buffer mode, and whether to continue or disable pipe operation at the end of transfer.

### EPNUM[3:0] bits (Endpoint Number)

The EPNUM[3:0] bits specify the endpoint number for the selected pipe. Setting 0000b indicates the pipe not used.

Set these bits so that the combination of the DIR and EPNUM[3:0] settings is different from those for other pipes. The EPNUM[3:0] bits can be set to 0000b for all pipes.

### DIR bit (Transfer Direction)

The DIR bit specifies the transfer direction for the selected pipe.

When the software sets this bit to 0, the USBFS uses the selected pipe for receiving. When the software sets this bit to 1, the USBFS uses the selected pipe for transmitting.

### SHTNAK bit (Pipe Disabled at End of Transfer)

The SHTNAK bit specifies whether to change the PIPEnCTR.PID[1:0] bits to 00b (NAK) at the end of transfer when the selected pipe is set in the receiving direction. The bit is valid for pipes 1 to 5 in the receiving direction.

When the software sets this bit to 1 for a receiving pipe, the USBFS changes the associated PIPEnCTR.PID[1:0] bits to 00b (NAK) on determining the transfer end. The USBFS determines that the transfer has ended on the following conditions:

- A short packet data (including a zero-length packet) was successfully received
- The transaction counter is used and the number of packets specified for the transaction counter are successfully received

### DBLB bit (Double Buffer Mode)

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe. The bit is valid for pipes 1 to 5.

### BFRE bit (BRDY Interrupt Operation Specification)

The BFRE bit specifies the BRDY interrupt generation timing from the USBFS to the CPU for the selected pipe.

When the software sets the BFRE bit to 1 and the selected pipe is in the receiving direction, the USBFS detects the transfer completion and generates the BRDY interrupt on reading the packet.

When a BRDY interrupt is generated with this setting, the software must write 1 to the BCLR bit in the port control register. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.

When the BFRE bit is set to 1 by software and the selected pipe is in the transmitting direction, the USBFS does not generate the BRDY interrupt. For details, see [section 29.3.3.1. BRDY interrupt](#).

### TYPE[1:0] bits (Transfer Type)

The TYPE[1:0] bits specify the transfer type for the pipe selected in the PIPESEL.PIPESEL[3:0] bits. Before setting PID to BUF and starting USB communication on the selected pipe, set the TYPE[1:0] bits to a value other than 00b.

### 29.2.33 PIPEMAXP : Pipe Maximum Packet Size Register

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x06C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DEVSEL[3:0]				—	—	—	MXPS[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	MXPS[8:0]	Maximum Packet Size <sup>*1</sup> <ul style="list-style-type: none"> <li>Pipes 1 and 2 1 byte (0x001) to 256 bytes (0x100) (Bit [9] not supported.)</li> <li>Pipes 3 to 5 8 bytes (0x008), 16 bytes (0x010), 32 bytes (0x020), 64 bytes (0x040) (Bits [9:7] and [2:0] not supported.)</li> <li>Pipes 6 to 9 1 byte (0x001) to 64 bytes (0x040) (Bits [9:7] not supported.)</li> </ul>	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
15:12	DEVSEL[3:0]	Device Select <sup>*2</sup> <ul style="list-style-type: none"> <li>0x0: Address 0000b</li> <li>0x1: Address 0001b</li> <li>0x2: Address 0010b</li> <li>0x3: Address 0011b</li> <li>0x4: Address 0100b</li> <li>0x5: Address 0101b</li> <li>Others: Setting prohibited</li> </ul>	R/W

Note: S-TYPE-3, P-TYPE-3

Note: The value of the MXPS[8:0] bits is 0x000 when no pipe is selected in the PIPESEL.PIPESEL[3:0] bits and 0x040 when a pipe is selected.

Note 1. Only set the MXPS[8:0] bits while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 2. Only set the DEVSEL[3:0] bits while PID is NAK. Before setting these bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00b (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEMAXP specifies the maximum packet size for pipes 1 to 9.

#### MXPS[8:0] bits (Maximum Packet Size)

The MXPS[8:0] bits specify the maximum data payload (maximum packet size) for the selected pipe.

Set these bits to the appropriate value for each transfer type based on the USB 2.0 specification. When MXPS[8:0] = 0, do not write to the FIFO buffer or set PID to BUF. These writes have no effect.

#### DEVSEL[3:0] bits (Device Select)

In host controller mode, the DEVSEL[3:0] bits specify the address of the target device for USB communication. Set up the device address in the associated DEVADDn (n = 0 to 5) register first, and then set these bits to the corresponding value. To set the DEVSEL[3:0] bits to 0x2, for example, first set the address in the DEVADD2 register.

In device controller mode, set these bits to 0x0.

### 29.2.34 PIPEPERI : Pipe Cycle Control Register

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x06E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	IITV[2:0] <sup>1</sup>	Interval Error Detection Interval Specifies the interval error detection timing for the selected pipe as the n-th power of 2 of the frame timing	R/W
11:3	—	These bits are read as 0. The write value should be 0.	R/W
12	IFIS	Isochronous IN Buffer Flush 0: Do not flush buffer 1: Flush buffer	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Only set the IITV[2:0] bits while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEPERI selects whether the buffer is flushed or not when an interval error occurred during isochronous IN transfers, and sets the interval error detection interval for pipes 1 to 9.

#### IITV[2:0] bits (Interval Error Detection Interval)

To change the IITV[2:0] bits to another value after they are set and USB communication is performed, set the PIPEnCTR.PID[1:0] bits to 00b (NAK) and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer.

The IITV[2:0] bits are not provided for pipes 3 to 5. Write 000b to bit positions of the IITV[2:0] bits associated with pipes 3 to 5.

#### IFIS bit (Isochronous IN Buffer Flush)

The IFIS bit specifies whether to flush the buffer when the pipe selected in the PIPESEL.PIPESEL[3:0] bits is used for isochronous IN transfers.

In device controller mode when the selected pipe is for isochronous IN transfers, the USBFS automatically clears the FIFO buffer if the USBFS fails to receive the IN token from the USB host within the interval set in the IITV[2:0] bits in terms of frames.

When double buffering is specified (PIPECFG.DBLB = 1), the USBFS only clears the data in the previously used plane.

The USBFS clears the FIFO buffer on receiving the SOF packet immediately after the frame in which the USBFS expected to receive the IN token. Even if the SOF packet is corrupted, the FIFO buffer is cleared at the time the SOF packet is expected to be received by using the internal interpolation function.

When the host controller function is selected, set this bit to 0.

Set this bit to 0 when the selected pipe is not for isochronous transfer.

### 29.2.35 PIPEnCTR : PIPEn Control Registers (n = 1 to 5)

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x070 + 0x2 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends buffer state) 1 0: STALL response 1 1: STALL response	R/W
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	Pipe Busy 0: Pipe n not in use for the transaction 1: Pipe n in use for the transaction	R
6	SQMON	Sequence Toggle Bit Confirmation 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set <sup>*2</sup> Sets the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA1	R/W <sup>*1</sup>
8	SQCLR	Sequence Toggle Bit Clear <sup>*2</sup> Clears the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	R/W <sup>*1</sup>
9	ACLRM	Auto Buffer Clear Mode <sup>*3</sup> 0: Disable 1: Enable (initialize all buffers)	R/W
10	ATREPM	Auto Response Mode <sup>*2</sup> 0: Disable auto response mode 1: Enable auto response mode	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	INBUFM	Transmit Buffer Monitor 0: No data to be transmitted is in the FIFO buffer 1: Data to be transmitted is in the FIFO buffer	R
15	BSTS	Buffer Status 0: Buffer access by the CPU disabled 1: Buffer access by the CPU enabled	R

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is read as 0.

Note 2. Only set the ATREPM bit or write 1 to the SQCLR or SQSET bit while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 3. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bit, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00 (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

PIPEnCTR can be set for any pipe selection in the PIPESEL register.

#### PID[1:0] bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction on the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. [Table 29.9](#) and [Table 29.10](#) show the basic operations of the USBFS (when there are no errors in the communication packets) based on the PID[1:0] bit setting.

After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the pipe has actually entered the NAK state. If the USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBFS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBFS sets PID to NAK on recognizing completion of the transfer when the selected pipe is in the receiving direction and the PIPECFG.SHTNAK bit for the selected pipe is set to 1 by software
- The USBFS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBFS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBFS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBFS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode

To specify the response type, set the PID[1:0] bits as follows:

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, set 00b (NAK) and then 01b (BUF)

**Table 29.9 Operation of the USBFS based on the PID[1:0] setting in host controller mode**

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
00b (NAK)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens
01b (BUF)	Bulk or interrupt	Does not depend on the setting	Issues tokens when the DVSTCTR0.UACT bit is 1 and the FIFO buffer associated with the selected pipe is ready for transmission and reception. Does not issue tokens when the DVSTCTR0.UACT bit is 0 or the FIFO buffer associated with the selected pipe is not ready for transmission or reception.
	Isochronous	Does not depend on the setting	Issues tokens regardless of the status of the FIFO buffer associated with the selected pipe.
10b (STALL) or 11b (STALL)	Does not depend on the setting	Does not depend on the setting	Does not issue tokens.

**Table 29.10 Operation of the USBFS based on the PID[1:0] setting in device controller mode (1 of 2)**

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
00b (NAK)	Bulk or interrupt	Does not depend on the setting	Returns NAK in response to the token from the USB host
	Isochronous	Does not depend on the setting	Returns nothing in response to the token from the USB host

**Table 29.10 Operation of the USBFS based on the PID[1:0] setting in device controller mode (2 of 2)**

PID[1:0] value	Transfer type	Transfer direction (DIR bit)	USBFS operation
01b (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception
	Bulk or interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the FIFO buffer associated with the selected pipe is ready for transmission. Otherwise, returns NAK.
	Isochronous	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer associated with the selected pipe is ready for reception. Otherwise, discards the data.
	Isochronous	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the associated FIFO buffer is ready for transmission. Otherwise, transmits a zero-length packet.
10b (STALL) or 11b (STALL)	Bulk or interrupt	Does not depend on the setting	Returns STALL in response to the token from the USB host
	Isochronous	Does not depend on the setting	Returns nothing in response to the token from the USB host

**PBUSY bit (Pipe Busy)**

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The USBFS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by software after PID is set to NAK allows you to check whether changing the pipe setting is possible. For details, see [section 29.3.4.1. Pipe control register switching procedures](#).

**SQMON bit (Sequence Toggle Bit Confirmation)**

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

When the selected pipe is not the isochronous transfer type, the USBFS toggles the SQMON flag on successful completion of the transaction. However, the USBFS does not toggle the SQMON flag when a DATA-PID mismatch occurs during transfer in the receiving direction.

**SQSET bit (Sequence Toggle Bit Set)**

Setting the SQSET bit to 1 through the software causes the USBFS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe. The USBFS clears the SQSET bit to 0.

**SQCLR bit (Sequence Toggle Bit Clear)**

Setting the SQCLR bit to 1 through the software causes the USBFS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0. The USBFS clears the SQCLR bit to 0.

**ACLRM bit (Auto Buffer Clear Mode)**

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

[Table 29.11](#) shows the data cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which this processing is required.

**Table 29.11 Data cleared by the USBFS when ACLRM = 1 (1 of 2)**

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe (two FIFO buffers in double buffer mode)	When initializing the selected pipe

**Table 29.11 Data cleared by the USBFS when ACLRM = 1 (2 of 2)**

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
2	Interval count value when the selected pipe is the isochronous transfer type	When resetting the interval count value
3	Internal flags related to the PIPECFG.BFRE bit	When changing the PIPECFG.BFRE setting
4	FIFO buffer toggle control	When changing the PIPECFG.DBLB setting
5	Internal flags related to the transaction count	When forcing the transaction count function to terminate

**ATREPM bit (Auto Response Mode)**

The ATREPM bit enables or disables auto response mode for the selected pipe.

This bit can be set to 1 in device controller mode when the selected pipe is the bulk transfer type. When the bit is set to 1, the USBFS responds to the token from the USB host as follows:

- When the selected pipe is set for bulk IN transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 1):
  - a. When the ATREPM bit = 1 and PID = BUF, the USBFS transmits a zero-length packet in response to the IN token.
  - b. The USBFS updates the sequence toggle bit (DATA-PID) each time the USBFS receives ACK from the USB host. In a single transaction, the IN token is received, a zero-length packet is transmitted, and then ACK is received. The USBFS does not generate the BRDY or BEMP interrupt.
- When the selected pipe is set for bulk OUT transfers (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 0):
 

When the ATREPM bit = 1 and PID = BUF, the USBFS returns NAK in response to the OUT token and generates an NRDY interrupt.

For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode. When the selected pipe uses isochronous transfer, always set this bit to 0.

In host controller mode, always set the ATREPM bit to 0.

**INBUFM bit (Transmit Buffer Monitor)**

The INBUFM bit indicates the FIFO buffer status for the selected pipe in the transmitting direction.

When the selected pipe is set in the transmitting direction (PIPECFG.DIR = 1), the USBFS sets this bit to 1 when the CPU or DMA/DTC completes writing data to at least one FIFO buffer plane.

The USBFS sets this bit to 0 when the USBFS completes transmission of the data from the FIFO buffer plane to which all the data is written. In double buffer mode (PIPECFG.DBLB = 1), the USBFS sets the INBUFM bit to 0 when the USBFS completes transmission of the data from the two FIFO buffer planes before the CPU or DMA/DTC completes writing data to one FIFO buffer plane.

The INBUFM bit indicates the same value as the BSTS bit when the selected pipe is in the receiving direction (PIPECFG.DIR = 0).

**BSTS bit (Buffer Status)**

The BSTS bit indicates the FIFO buffer status for the selected pipe.

The meaning of the BSTS bit depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in [Table 29.12](#).



**Table 29.12** BSTS bit operation

DIR value	BFRE value	DCLRM value	BSTS bit function
0	0	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
		1	Setting prohibited
	1	0	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 when the software sets the BCLR bit in the port control register to 1 after the data read is complete
		1	Sets to 1 when receive data can be read from the FIFO buffer, and clears to 0 on completion of data read
1	0	0	Sets to 1 when transmit data can be written to the FIFO buffer, and clears to 0 on completion of data write
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

### 29.2.36 PIPEnCTR : PIPEn Control Registers (n = 6 to 9)

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x07A + 0x2 × (n - 6)

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	BSTS	—	—	—	—	—	ACL M	SQCL R	SQSE T	SQM ON	PBUS Y	—	—	—	PID[1:0]
------------	------	---	---	---	---	---	----------	-----------	-----------	-----------	-----------	---	---	---	----------

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID 0 0: NAK response 0 1: BUF response (depends buffer state) 1 0: STALL response 1 1: STALL response	R/W
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	Pipe Busy 0: Pipe n not in use for the transaction 1: Pipe n in use for the transaction	R
6	SQMON	Sequence Toggle Bit Confirmation 0: DATA0 1: DATA1	R
7	SQSET	Sequence Toggle Bit Set* <sup>1</sup> Sets the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Set the expected value for the next transaction to DATA0	W
8	SQCLR	Sequence Toggle Bit Clear* <sup>1</sup> Clears the sequence toggle bit for pipe n. 0: Invalid (writing 0 has no effect) 1: Clear the expected value for the next transaction to DATA0	W
9	ACLRM	Auto Buffer Clear Mode* <sup>2</sup> 0: Disable 1: Enable (all buffers initialized)	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	BSTS	Buffer Status 0: Buffer access disabled 1: Buffer access enabled	R

Note: S-TYPE-3, P-TYPE-3

Note 1. Only write 1 to the SQCLR or SQSET bit while PID is NAK. Before setting these bits, check that the PBUSY bit is 0, and then change the PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00b (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

Note 2. Only set the ACLRM bit while PID is NAK and before the pipe is selected in the CURPIPE[3:0] bits in the port select register. Before setting this bits, check that the PIPEnCTR.PBUSY bit is 0, and then change the PIPEnCTR.PID[1:0] bits from 01b (BUF) to 00b (NAK). If the PID[1:0] bits are changed to 00b (NAK) by the USBFS, checking the PBUSY bit through the software is not necessary.

### PID[1:0] bits (Response PID)

The PID[1:0]bits specify the response type for the next transaction of the selected pipe.

The default PID[1:0] setting is NAK. Change the PID[1:0] setting to BUF to use the associated pipe for USB transfer. [Table 29.9](#) and [Table 29.10](#) show the basic operation (when there are no errors in the transmitted and received packets) of the USBFS depending on the PID[1:0] setting.

After changing the PID[1:0] setting from BUF to NAK through the software during USB communication on the selected pipe, check that the PBUSY bit is 1 to see if USB transfer on the selected pipe has actually entered the NAK state. If the USBFS changes the PID[1:0] bits to NAK, checking the PBUSY bit through the software is not necessary.

The USBFS changes the PIPEnCTR.PID[1:0] setting in the following cases:

- The USBFS sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the selected pipe
- The USBFS sets PID to NAK on detecting a USB bus reset in device controller mode
- The USBFS sets PID to NAK on detecting a reception error, such as a CRC error, three consecutive times in host controller mode
- The USBFS sets PID to STALL (11b) on receiving the STALL handshake in host controller mode

To specify each response type, set the PID[1:0] bits as follows:

- To transition from NAK (00b) to STALL, set 10b
- To transition from BUF (01b) to STALL, set 11b
- To transition from STALL (11b) to NAK, set 10b and then 00b
- To transition from STALL to BUF, set 00b (NAK) and then 01b (BUF)

### PBUSY bit (Pipe Busy)

The PBUSY bit indicates whether the selected pipe is being used for the current transaction.

The USBFS changes the PBUSY bit from 0 to 1 on start of the USB transaction for the selected pipe, and changes the PBUSY bit from 1 to 0 on completion of one transaction.

Reading the PBUSY bit by software after PID is set to NAK allows you to check whether changing the pipe setting is possible.

### SQMON bit (Sequence Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the selected pipe.

The USBFS toggles the SQMON bit on successful completion of the transaction. However, the USBFS does not toggle the SQMON bit when a DATA-PID mismatch occurs during transfer in the receiving direction.

### SQSET bit (Sequence Toggle Bit Set)

Setting the SQSET bit to 1 through the software causes the USBFS to set DATA1 as the expected value of the sequence toggle bit for the next transaction on the selected pipe. The USBFS sets the SQSET bit to 0.

### SQCLR bit (Sequence Toggle Bit Clear)

Setting the SQCLR bit to 1 through the software causes the USBFS to clear the expected value of the sequence toggle bit for the next transaction on the selected pipe to DATA0. The USBFS sets the SQCLR bit to 0.

### ACLRM bit (Auto Buffer Clear Mode)

The ACLRM bit enables or disables auto buffer clear mode for the selected pipe. To completely clear the data in the FIFO buffer allocated to the selected pipe, write 1 and then 0 to the ACLRM bit continuously.

Table 29.13 shows the data cleared by writing 1 and 0 continuously to the ACLRM bit and the cases in which this processing is required.

**Table 29.13 Data cleared by the USBFS when ACLRM = 1**

Number	Data cleared by setting the ACLRM bit	Situations requiring data clear
1	All data in the FIFO buffer allocated to the selected pipe	When initializing the selected pipe
2	Interval count value when the selected pipe is the isochronous transfer type	When resetting the interval count value
3	Internal flags related to the PIPECFG.BFRE bit	When changing the PIPECFG.BFRE setting
4	Internal flags related to the transaction count	When forcing the transaction count function to terminate

**BSTS bit (Buffer Status)**

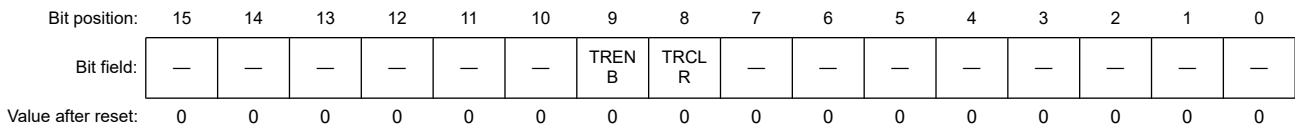
The BSTS bit indicates the FIFO buffer status for the selected pipe.

The meaning of the BSTS bit depends on the PIPECFG.DIR, PIPECFG.BFRE, and DnFIFOSEL.DCLRM settings, as shown in Table 29.12.

**29.2.37 PIPEnTRE : PIPEn Transaction Counter Enable Register (n = 1 to 5)**

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x090 + 0x4 × (n - 1)



Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	TRCLR	Transaction Counter Clear 0: Invalid (writing 0 has no effect) 1: Clear counter value	R/W
9	TRENB	Transaction Counter Enable 0: Disable transaction counter 1: Enable transaction counter	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Set each bit in PIPEnTRE while PID is NAK. Before setting these bits after changing the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits are changed to NAK by the USBFS, checking the PBUSY bit through the software is not necessary.

**TRCLR bit (Transaction Counter Clear)**

When the TRCLR bit sets to 1, the USBFS clears the value of the transaction counter associated with the selected pipe and then sets the TRCLR bit to 0.

**TRENB bit (Transaction Counter Enable)**

The TRENB bit enables or disables the transaction counter.

For receiving pipes, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN.TRNCNT[15:0] bits through the software allows the USBFS to control hardware on having received the number of packets equal to the TRNCNT[15:0] setting, as follows:

- When the PIPECFG.SHTNAK bit is 1, the USBFS changes the PID bits to NAK for the associated pipe on having received the number of packets equal to the TRNCNT[15:0] setting
- When the PIPECFG.BFRE bit is 1, the USBFS asserts the BRDY interrupt on having received the number of packets equal to the TRNCNT[15:0] setting and then reading the last received data

For transmitting pipes, set the TRENB bit to 0.

When the transaction counter is not used, set this bit to 0. When the transaction counter is used, set the TRNCNT[15:0] bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.

### 29.2.38 PIPEnTRN : PIPEn Transaction Counter Register (n = 1 to 5)

Base address: USBFS = 0x4025\_0000  
USBFS\_NS = 0x5025\_0000

Offset address: 0x092 + 0x4 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRNCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TRNCNT[15:0]	Transaction Counter When written to, this bit specifies the total packets (number of transactions) to be received by the selected pipe. When read from, when PIPEnTRE.TRENB is 0, this bit indicates the specified number of transactions. When PIPEnTRE.TRENB is 1, this bit indicates the current transaction count.	R/W

Note: S-TYPE-3, P-TYPE-3

The PIPEnTRN registers retain their settings during a USB bus reset.

#### TRNCNT[15:0] bits (Transaction Counter)

The USBFS increments the value of the TRNCNT[15:0] bits by 1 when all of the following conditions are satisfied on receiving the packet:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value ≠ current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[9:0] setting

The USBFS clears the value of the TRNCNT[15:0] bits to 0 when any of the following conditions are satisfied:

All of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- (TRNCNT[15:0] set value = current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the PIPEMAXP.MXPS[9:0] setting

Both of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The USBFS received a short packet

Both of the following conditions are satisfied:

- The PIPEnTRE.TRENB bit = 1
- The PIPEnTRE.TRCLR bit was set to 1 by software

For transmitting pipes, set the TRNCNT[15:0] bits to 0. When the transaction counter is not used, set the TRNCNT[15:0] bits to 0.

Setting the number of transactions to be transferred to the TRNCNT[15:0] bits is only enabled when the PIPEnTRE.TRENB bit is 0. To set the number of transactions to be transferred, set the TRCLR bit to 1 to clear the current counter value before setting the PIPEnTRE.TRENB bit to 1.

### 29.2.39 DEVADDn : Device Address n Configuration Register (n = 0 to 5)

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x0D0 + 0x2 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	USBSPD[1:0]	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
7:6	USBSPD[1:0]	Transfer Speed of Communication Target Device 0 0: Do not use DEVADDn 0 1: Low-speed 1 0: Full-speed 1 1: Setting prohibited	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The DEVADDn register specifies the transfer speed of the peripheral device that is the communication target for pipes 0 to 9.

In host controller mode, set all DEVADDn bits before starting communication to any pipes. Only change the bits in DEVADDn when no valid pipes are using the bit settings. A valid pipe is defined as one that satisfies both of the following conditions:

- The target device of the DEVADDn register is selected in the DEVSEL[3:0] bits
- The PID[1:0] bits are set to BUF for the selected pipe, or the selected pipe is the DCP with the DCPCTR.SUREQ bit set to 1

In device controller mode, set all bits in this register to 0.

#### USBSPD[1:0] bits (Transfer Speed of Communication Target Device)

The USBSPD[1:0] bits specify the USB transfer speed of the target peripheral device. Set these bits to 10b when a full-speed device is connected through the hub. In host controller mode, the USBFS generates packets based on the USBSPD[1:0] setting. In device controller mode, set these bits to 00b.

### 29.2.40 DPUSR0R : Deep Software Standby USB Transceiver Control/Pin Monitor Register

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x400

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DVBS TS0	—	DOVC B0	DOVC A0	—	—	DM0	DP0
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	x	0	0	x	x
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	FIXPH Y0	DRPD 0	—	RPUE 0	SRPC 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SRPC0* <sup>1</sup>	USB Single-ended Receiver Control 0: Disable input through DP and DM inputs 1: Enable input through DP and DM inputs	R/W
1	RPUE0* <sup>1</sup>	DP Pull-Up Resistor Control 0: Disable DP pull-up resistor 1: Enable DP pull-up resistor	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	DRPD0* <sup>1</sup>	D+/D- Pull-Down Resistor Control 0: Disable DP/DM pull-down resistor 1: Enable DP/DM pull-down resistor	R/W
4	FIXPHY0	USB Transceiver Output Fix 0: Fix outputs in Normal mode and on return from Deep Software Standby mode 1 1: Fix outputs on transition to Deep Software Standby mode 1	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
16	DP0	USB D+ Input Indicates D+ input signal on the USBFS side	R
17	DM0	USB D- Input Indicates D- input signal on the USBFS side	R
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	DOVCA0	USB OVRCURA-DS Input* <sup>2</sup> Indicates OVRCURA-DS input signal on the USBFS side	R
21	DOVCB0	USB OVRCURB-DS Input* <sup>2</sup> Indicates OVRCURB-DS input signal on the USBFS side	R
22	—	The read value is undefined. The write value should be 0.	R/W
23	DVBST0	USB VBUS Input Indicates VBUS input signal on the USBFS side	R
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: This register is accessed by the clock of PCLKB/64 to reduce current consumption in Deep Software Standby mode1.

Note 1. Use this bit during operation in Deep Software Standby mode 1. For details, see [section 29.3.1.5. Release from Deep Software Standby mode1 because of USB suspend/resume interrupts.](#)

Note 2. OVRCURA or OVRCURB can not be used. Only OVRCURA-DS or OVRCURB-DS can be used in Deep software standby mode1.

### SRPC0 bit (USB Single-ended Receiver Control)

The SRPC0 bit controls the D+ and D- inputs of the USB transceiver. In host controller mode, set this bit to 1. In device controller mode, set this bit to 0 when disconnected, set to 1 when suspended. This bit is only valid when the FIXPHY0 bit is 1.

### FIXPHY0 bit (USB Transceiver Output Fix)

The FIXPHY0 bit keeps the outputs of the USB transceiver disabled.

## 29.2.41 DPUSR1R : Deep Software Standby USB Suspend/Resume Interrupt Register

Base address: USBFS = 0x4025\_0000  
 USBFS\_NS = 0x5025\_0000

Offset address: 0x404

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DVBIN T0	—	DOVR CRB0	DOVR CRA0	—	—	DMINT 0	DPINT 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DVBS E0	—	DOVR CRBE 0	DOVR CRAE 0	—	—	DMINT E0	DPINT E0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DPINTE0	USB DP Interrupt Enable/Clear 0: Disable recovery from Deep Software Standby mode 1 by DP input 1: Enable recovery from Deep Software Standby mode 1 by DP input	R/W
1	DMINTE0	USB DM Interrupt Enable/Clear 0: Disable recovery from Deep Software Standby mode 1 by DM input 1: Enable recovery from Deep Software Standby mode 1 by DM input	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	DOVRCRAE0	USB OVRCURA-DS Interrupt Enable/Clear*1 0: Disable recovery from Deep Software Standby mode 1 by OVRCURA-DS input 1: Enable recovery from Deep Software Standby mode 1 by OVRCURA-DS input	R/W
5	DOVRCRBE0	USB OVRCURB-DS Interrupt Enable/Clear*1 0: Disable recovery from Deep Software Standby mode 1 by OVRCURB-DS input 1: Enable recovery from Deep Software Standby mode 1 by OVRCURB-DS input	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	DVBSE0	USB VBUS Interrupt Enable/Clear 0: Disable recovery from Deep Software Standby mode 1 by VBUS input 1: Enable recovery from Deep Software Standby mode 1 by VBUS input	R/W
15:8	—	These bits are read as 0. The write value should be 0..	R/W
16	DPINT0	USB DP Interrupt Source Recovery 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of DP	R
17	DMINT0	USB DM Interrupt Source Recovery 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of DM input	R
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	DOVRCRA0	USB OVRCURA-DS Interrupt Source Recovery*1 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of OVRCURA-DS input	R
21	DOVRCRB0	USB OVRCURB-DS Interrupt Source Recovery*1 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of OVRCURB-DS input	R
22	—	This bit is read as 0. The write value should be 0.	R/W
23	DVBINT0	USB VBUS Interrupt Source Recovery 0: System has not recovered from Deep Software Standby mode 1 1: System recovered from Deep Software Standby mode 1 because of VBUS input	R
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: This register is accessed by the clock of PCLKB/64 to reduce current consumption in Deep Software Standby mode1.

Note 1. OVRCURA or OVRCURB can not be used. Only OVRCURA-DS or OVRCURB-DS can be used in Deep Software Standby mode1.

#### **DPINTE0 bit (USB DP Interrupt Enable/Clear)**

The DPINTE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the DP input of the USBFS. Writing 0 to this bit while the DPINT0 bit is 1 sets the DPINT0 bit to 0.

#### **DMINTE0 bit (USB DM Interrupt Enable/Clear)**

The DMINTE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the DM input of the USBFS. Writing 0 to this bit while the DMINT0 bit is 1 clears the DMINT0 bit to 0.

#### **DOVRCRAE0 bit (USB OVRCURA-DS Interrupt Enable/Clear)**

The DOVRCRAE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the OVRCURA-DS input of the USBFS. Writing 0 to this bit while the DOVRCRA0 bit is 1 clears the DOVRCRA0 bit to 0.

#### **DOVRCRBE0 bit (USB OVRCURB-DS Interrupt Enable/Clear)**

The DOVRCRBE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the OVRCURB-DS input of the USBFS. Writing 0 to this bit while the DOVRCRB0 bit is 1 clears the DOVRCRB0 bit to 0.

#### **DVBSE0 bit (USB VBUS Interrupt Enable/Clear)**

The DVBSE0 bit enables or disables triggering of recovery from Deep Software Standby mode 1 by the VBUS input of the USBFS. Writing 0 to this bit while the DVBINT0 bit is 1 clears the DVBINT0 bit to 0.

#### **DPINT0 bit (USB DP Interrupt Source Recovery)**

The DPINT0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the DP input of the USBFS. This recovery is only enabled when the DPINTE0 bit is 1. Writing 0 to the DPINTE0 bit while this bit is 1 clears this bit to 0.

#### **DMINT0 bit (USB DM Interrupt Source Recovery)**

The DMINT0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the DM input of the USBFS. This recovery is only enabled when the DMINTE0 bit is 1. Writing 0 to the DMINTE0 bit while this bit is 1 clears this bit to 0.

#### **DOVRCRA0 bit (USB OVRCURA-DS Interrupt Source Recovery)**

The DOVRCRA0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the OVRCURA-DS input of the USBFS. This recovery is only enabled when the DOVRCRAE0 bit is 1. Writing 0 to the DOVRCRAE0 bit while this bit is 1 clears this bit to 0.

#### **DOVRCRB0 bit (USB OVRCURB-DS Interrupt Source Recovery)**

The DOVRCRB0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the OVRCURB-DS input of the USBFS. This recovery is only enabled when the DOVRCRBE0 bit is 1. Writing 0 to the DOVRCRBE0 bit while this bit is 1 clears this bit to 0.

#### **DVBINT0 bit (USB VBUS Interrupt Source Recovery)**

The DVBINT0 bit indicates that the system has returned from Deep Software Standby mode 1 because of the VBUS input of the USBFS. This recovery is only enabled when the DVBSE0 bit is 1. Writing 0 to the DVBSE0 bit while this bit is 1 clears this bit to 0.

## 29.3 Operation

### 29.3.1 System Control

This section describes register settings required for initializing the USBFS and controlling power consumption.



### 29.3.1.1 Setting data to the USBFS registers

Setting the SYSCFG.USBE bit to 1 after starting the clock supply (SYSCFG.SCKE bit = 1) enables and starts USBFS operation.

### 29.3.1.2 Selecting the controller function

The USBFS can operate as either a host or device controller.

Use the SYSCFG.DCFM bit to select one of these USBFS functions. The DCFM bit must be changed in the initial settings immediately after a reset or in the D+ pull-up-disabled state (SYSCFG.DPRPU bit = 0) and D+ and D- pull-down-disabled state (SYSCFG.DRPD bit = 0).

### 29.3.1.3 Controlling the USB data bus using resistors

The USBFS provides pull-up and pull-down resistors for the D+ and D- lines. Pull these lines up or down by setting the SYSCFG.DPRPU and DRPD bits.

In device controller mode, confirm that connection to the USB host is made, and then set the SYSCFG.DPRPU bit to 1 and pull up the D+ line (in full-speed communication).

When the SYSCFG.DPRPU bit is set to 0 during communication with a PC, the USBFS disables the pull-up resistor of the USB data line, thereby notifying the USB host of disconnection.

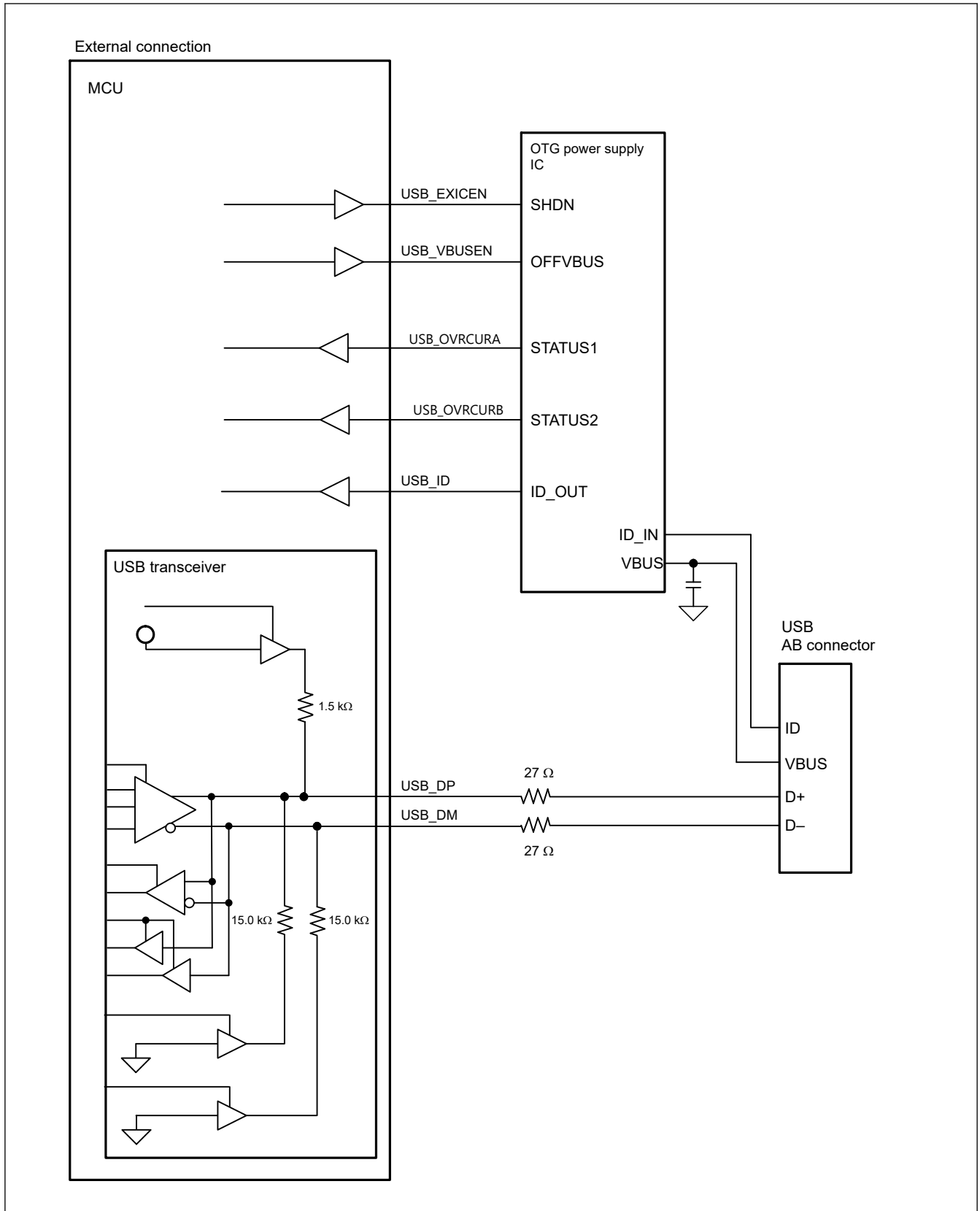
In host controller mode, set the SYSCFG.DRPD bit to 1 to pull down the D+ and D- lines.

**Table 29.14 USB data bus resistor control**

SYSCFG register settings		USB data bus control		Function
DRPD bit	DPRPU bit	D-	D+	
0	0	Open	Open	When resistors not used
0	1	Open	Pull-up	When operating as a device controller at full-speed
1	0	Pull-down	Pull-down	When operating as a host controller
1	1	—	—	Setting prohibited

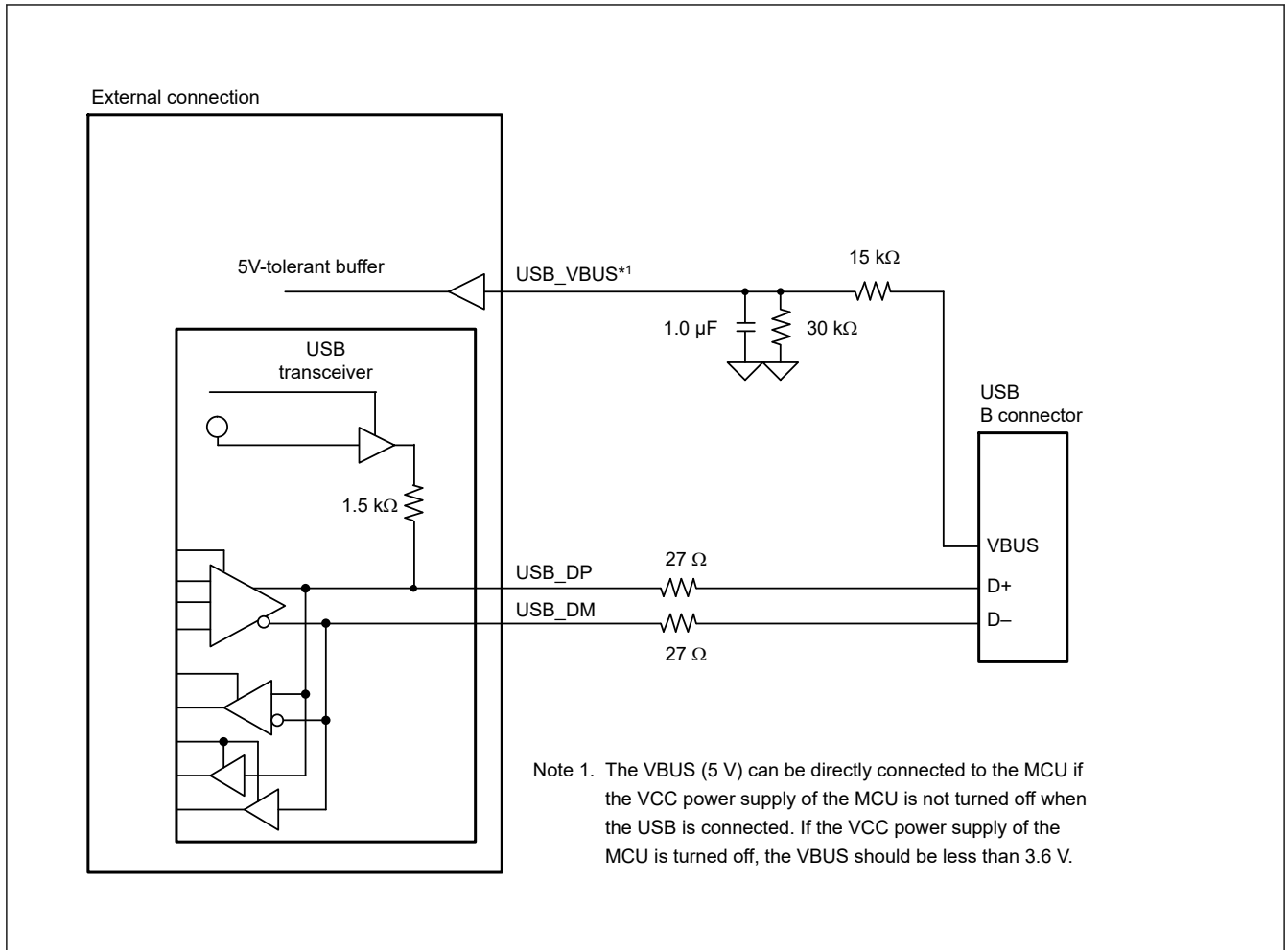
### 29.3.1.4 Example external connection circuits

[Figure 29.2](#) shows an example OTG connection in the self-powered system. The USBFS controls the pull-up resistor of the D+ line and the pull-down resistor of D+ and D- lines. Select pull-up and pull-down for the lines in the SYSCFG.DPRPU and SYSCFG.DRPD bits. In device controller mode, the pull-up resistor of USB data line is disabled if SYSCFG.DPRPU bit is set to 0 while communicating with the USB host. The USBFS can use this to notify the USB host of a device disconnect.



**Figure 29.2 Example OTG connection in a self-powered system**

Figure 29.3 shows an example device connection in a self-powered system.



**Figure 29.3 Example device connection in a self-powered system**

Figure 29.4 shows an example host connection.

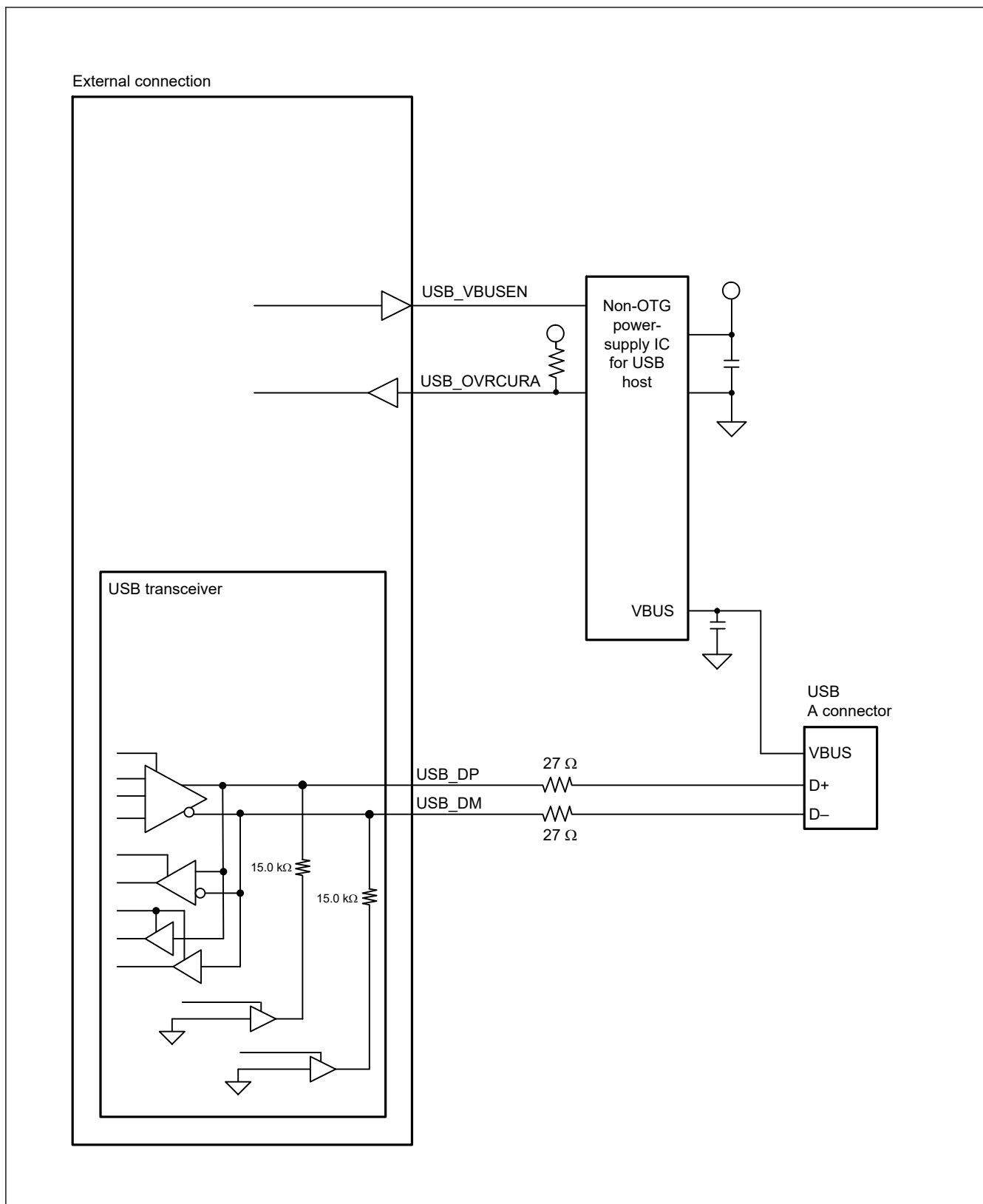
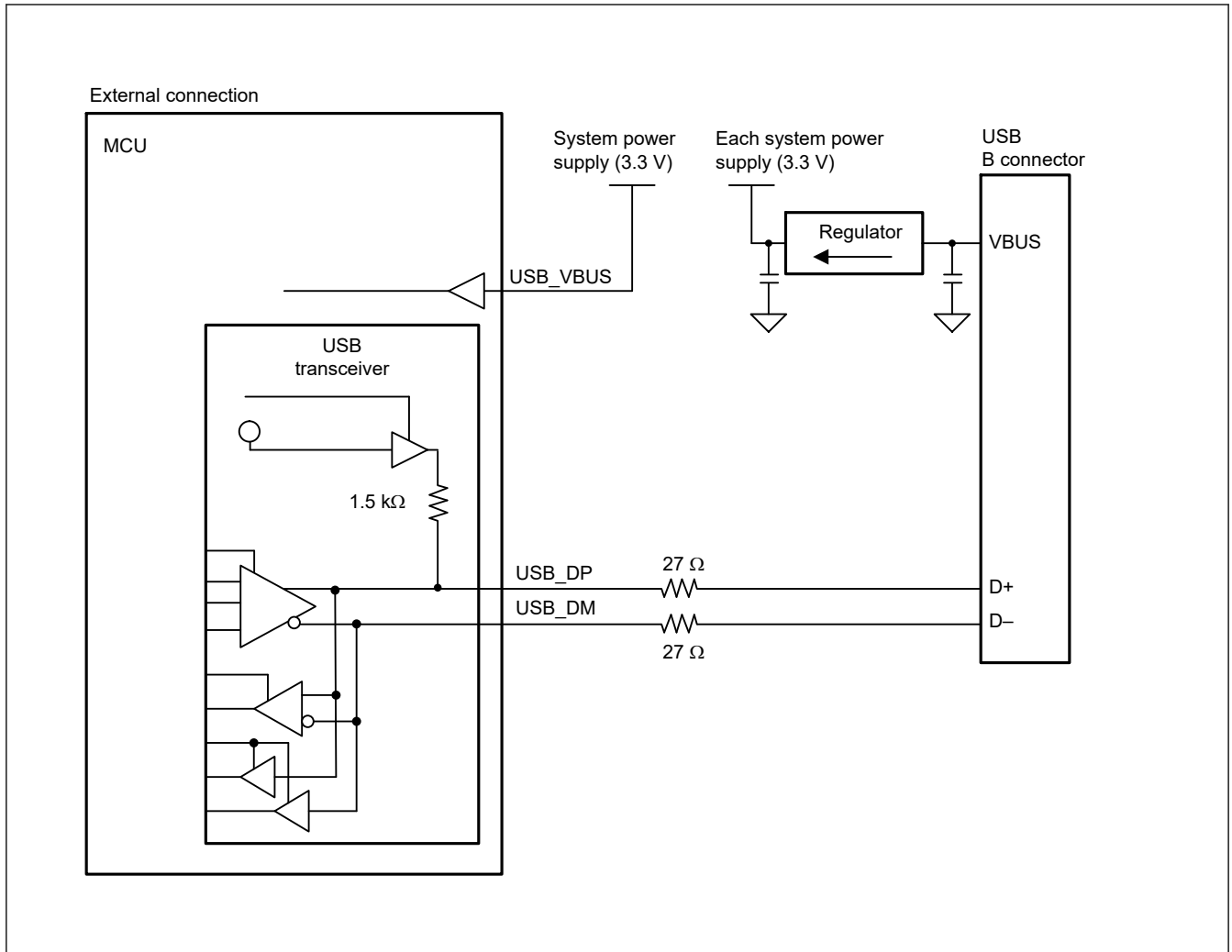


Figure 29.4 Example host connection

Figure 29.5 shows an example device connection in a bus-powered system.



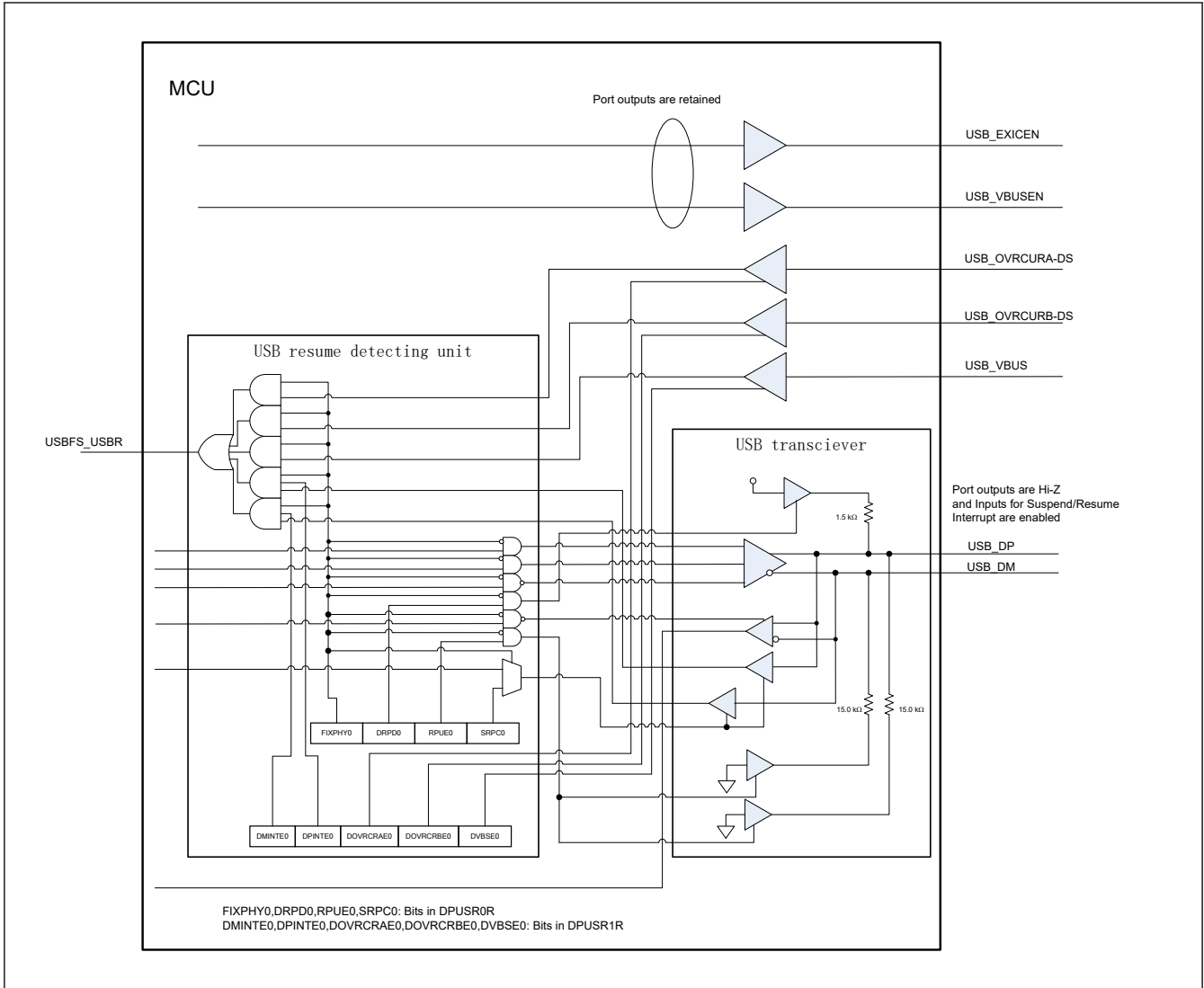
**Figure 29.5 Example device connection in a bus-powered state**

The examples of external circuits given in this section are simplified circuits, and their operation in every system is not guaranteed.

### 29.3.1.5 Release from Deep Software Standby mode1 because of USB suspend/resume interrupts

Deep Software Standby mode1 can be canceled by a USB suspend/resume interrupt. USB suspend/resume interrupts are detected by the USB resume detecting unit, which controls and monitors the USB I/O pins to detect the interrupts.

Figure 29.6 shows a schematic diagram of the connection between the USB resume detecting unit and the USB I/O pins.



**Figure 29.6 Connection between the USB resume detecting unit and the USB I/O pins**

Table 29.15 shows the USB suspend and resume interrupt sources and their associated I/O pins.

**Table 29.15 USB suspend and resume interrupt sources and their associated I/O pins**

USB operating mode	Source	Pin name
Device, OTG	Resume	USB_DP
Host, OTG	Attach or detach	USB_DP, USB_DM
Device	Attach or detach	USB_VBUS
Host	Overcurrent detection	USB_OVRCURA-DS
OTG	Overcurrent detection	USB_OVRCURA-DS, USB_OVRCURB-DS

Figure 29.7 shows the flow for setting the USBFS when entering Deep Software Standby mode 1 from either host or device controller mode. Figure 29.8 shows the flow for setting the USBFS when canceling Deep Software Standby mode 1 from host controller mode. Figure 29.9 shows the flow for setting the USBFS when canceling Deep Software Standby mode 1 from device controller mode.

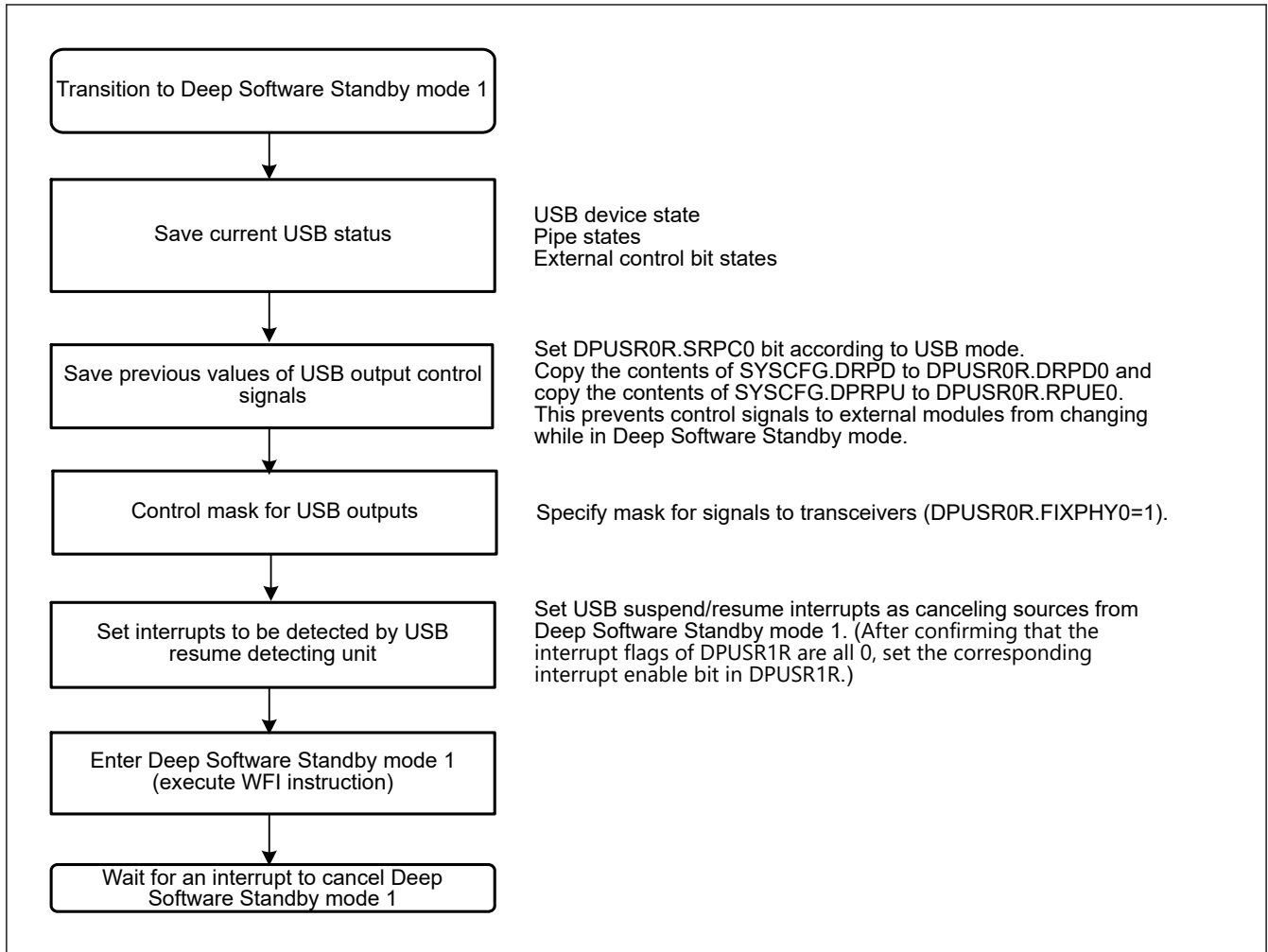


Figure 29.7 USBFS setup flow for transition to Deep Software Standby mode 1 as host or device controller

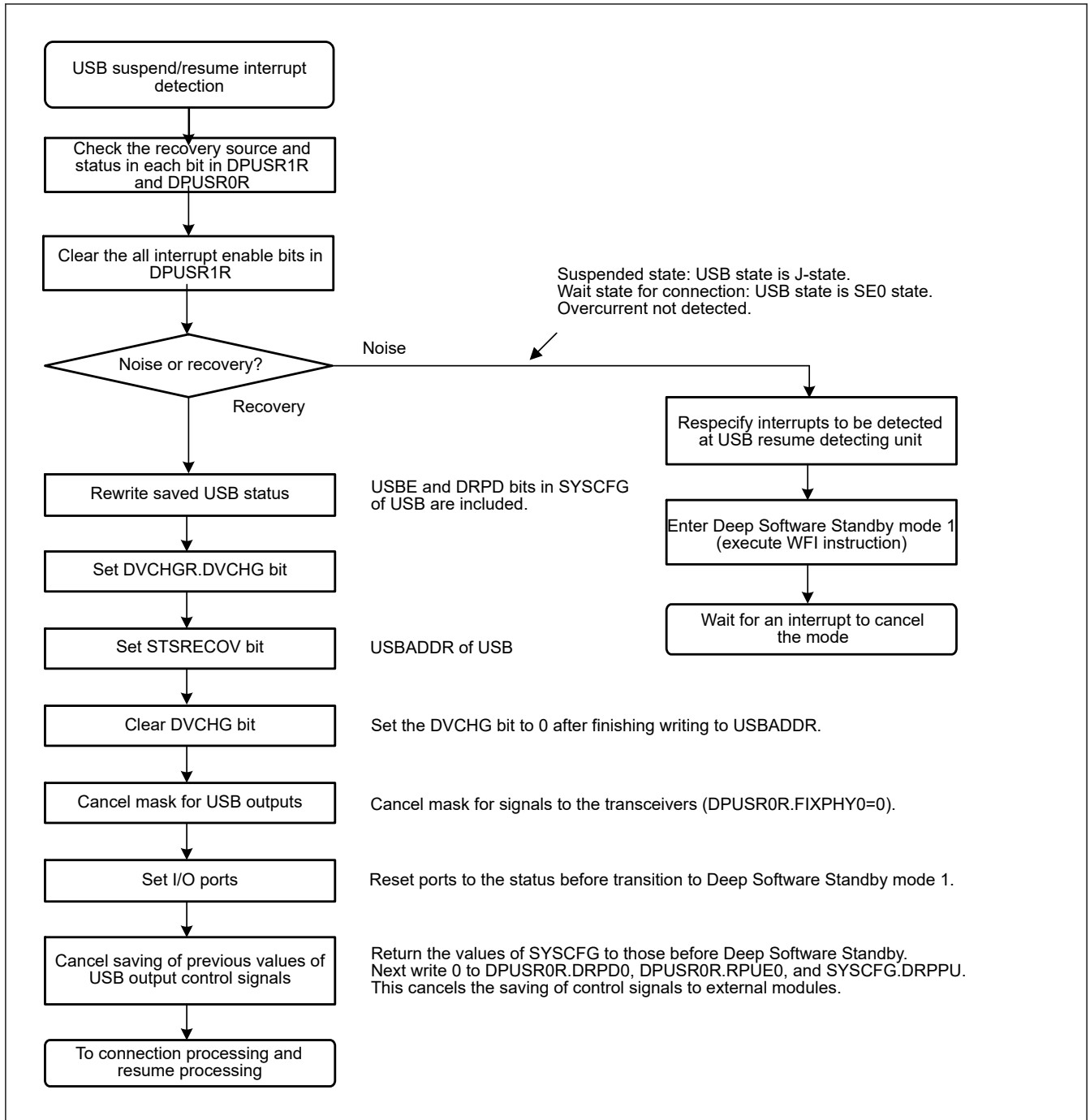


Figure 29.8 USBFS setup flow for canceling Deep Software Standby mode 1 as host controller



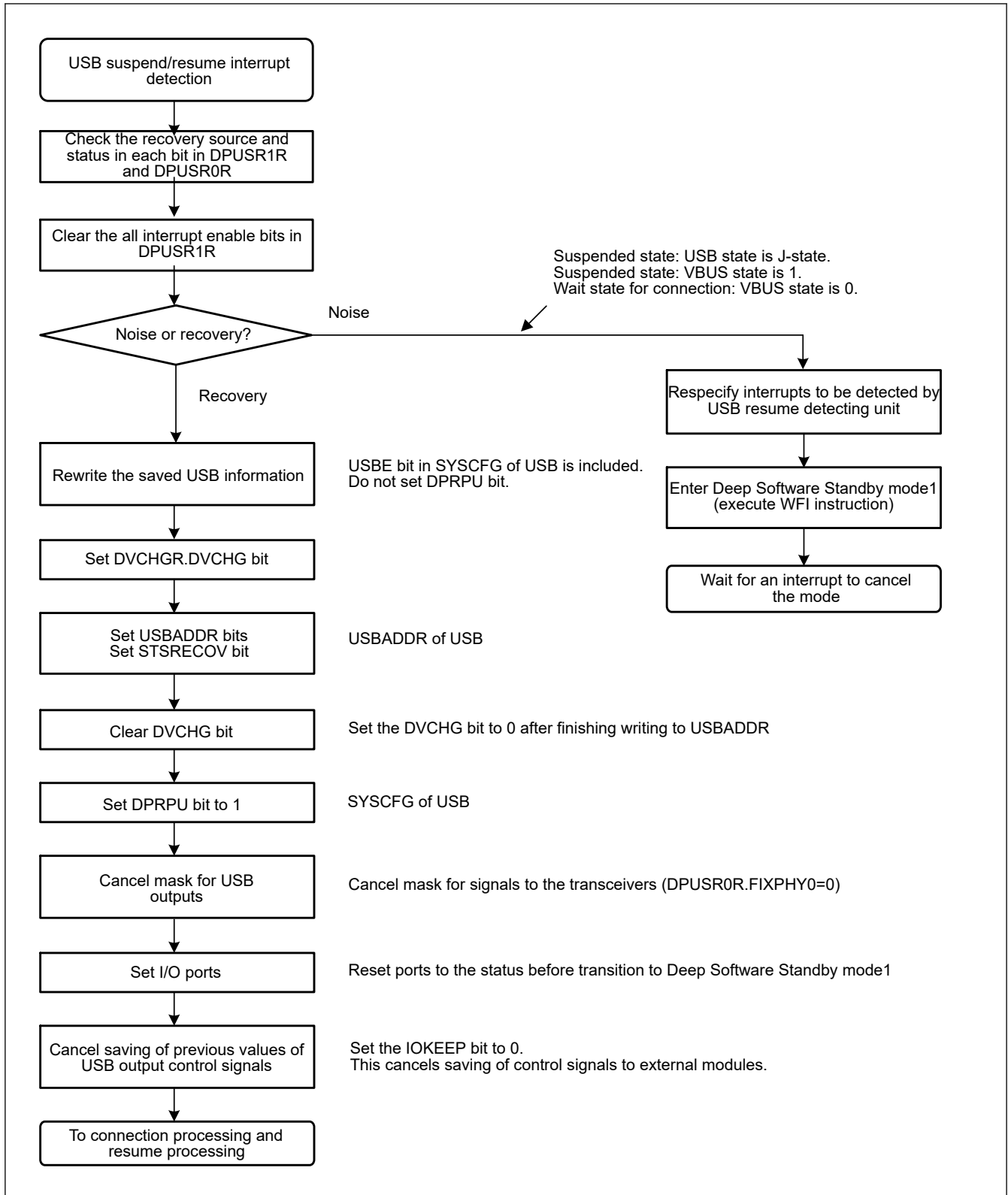


Figure 29.9 USBFS setup flow for canceling Deep Software Standby mode 1 as device controller

### 29.3.2 Interrupts

Table 29.16 lists the interrupt sources in the USBFS. When an interrupt generation condition is satisfied and the interrupt output is enabled using the associated interrupt enable register, a USBFS interrupt request is issued to the Interrupt Controller Unit (ICU) and an USBFS interrupt is generated.

Table 29.16 Interrupt sources (1 of 2)

Bit to be set to 1	Name	Interrupt source	Applicable controller function	Status flag
VBINT	VBUS interrupt	<ul style="list-style-type: none"> <li>A change in the state of the USB_VBUS input pin was detected (low to high or high to low)</li> </ul>	Host or device* <sup>1</sup>	INTSTS0.VBSTS
RESM	Resume interrupt	<ul style="list-style-type: none"> <li>A change in the state of the USB bus was detected in the Suspend state (J-state to K-state or J-state to SE0)</li> </ul>	Device	—
SOFR	Frame number update interrupt	<p>In host controller mode:</p> <ul style="list-style-type: none"> <li>An SOF packet with a different frame number was transmitted</li> </ul> <p>In device controller mode:</p> <ul style="list-style-type: none"> <li>An SOF packet with a different frame number was received</li> </ul>	Host or device	—
DVST	Device state transition interrupt	<ul style="list-style-type: none"> <li>One of the following device state transitions was detected: <ul style="list-style-type: none"> <li>USB bus reset was detected</li> <li>Suspend state was detected</li> <li>SET_ADDRESS request was received</li> <li>SET_CONFIGURATION request was received</li> </ul> </li> </ul>	Device	INTSTS0.DVSQ[2:0]
CTRT	Control transfer stage transition interrupt	<ul style="list-style-type: none"> <li>A control transfer stage transition was detected because of one of the following: <ul style="list-style-type: none"> <li>Setup stage completed</li> <li>Control write transfer status stage transition occurred</li> <li>Control read transfer status stage transition occurred</li> <li>Control transfer completed</li> <li>Control transfer sequence error occurred</li> </ul> </li> </ul>	Device	INTSTS0.CTSQ[2:0]
BEMP	Buffer empty interrupt	<ul style="list-style-type: none"> <li>The buffer is empty after all FIFO buffer data was transmitted</li> <li>A packet larger than the maximum packet size was received</li> </ul>	Host or device	BEMPSTS.PIPEnBEMP
NRDY	Buffer not ready interrupt	<p>In host controller mode</p> <ul style="list-style-type: none"> <li>A STALL response was received from the peripheral device in response to the issued token</li> <li>The response from the peripheral device in response to the issued token was not received successfully (no response three times consecutively or packet reception error three times consecutively)</li> <li>An overrun or underrun error occurred during isochronous transfer</li> </ul> <p>In device controller mode</p> <ul style="list-style-type: none"> <li>NAK was returned for an IN or OUT token while the PID[1:0] bits were set to 01b (BUF)</li> <li>A CRC error or bit stuffing error occurred during data reception in isochronous transfer</li> <li>An overrun or underrun occurred during data reception in isochronous transfer</li> </ul>	Host or device	NRDYSTS.PIPEnNRDY
BRDY	Buffer ready interrupt	<ul style="list-style-type: none"> <li>The buffer is ready (readable or writable state)</li> </ul>	Host or device	BRDYSTS.PIPEnBRDY
OVRRCR	Overcurrent input change interrupt	<ul style="list-style-type: none"> <li>USB_OVRCURA, USB_OVRCURA-DS, USB_OVRCURB or USB_OVRCURB-DS input pin state change was detected (low to high or high to low)</li> </ul>	Host	INTSTS1.OVRRCR
BCHG	Bus change interrupt	<ul style="list-style-type: none"> <li>USB bus state change was detected</li> </ul>	Host or device	SYSSTS0.LNST[1:0]
DTCH	Disconnect detection during full-speed operation	Peripheral device disconnect was detected in full-speed operation	Host	DVSTCTR0.RHST[2:0]
ATTCH	Device connect detection interrupt	<ul style="list-style-type: none"> <li>J-state or K-state was detected on the USB bus for 2.5 <math>\mu</math>s continuously</li> </ul> <p>This interrupt can be used to check whether peripheral devices are connected</p>	Host	—

**Table 29.16** Interrupt sources (2 of 2)

Bit to be set to 1	Name	Interrupt source	Applicable controller function	Status flag
EOFERR	EOF error detection interrupt	<ul style="list-style-type: none"> <li>An EOF error was detected for a peripheral device</li> </ul>	Host	—
SACK	Setup normal interrupt	<ul style="list-style-type: none"> <li>A setup transaction normal response (ACK) was received</li> </ul>	Host	—
SIGN	Setup error interrupt	<ul style="list-style-type: none"> <li>A setup transaction error (no response or ACK packet corruption) was detected three consecutive times</li> </ul>	Host	—

Note 1. Although this interrupt can be generated in host controller mode, it is not usually used in this mode.

Figure 29.10 shows the circuits related to the USBFS interrupts.

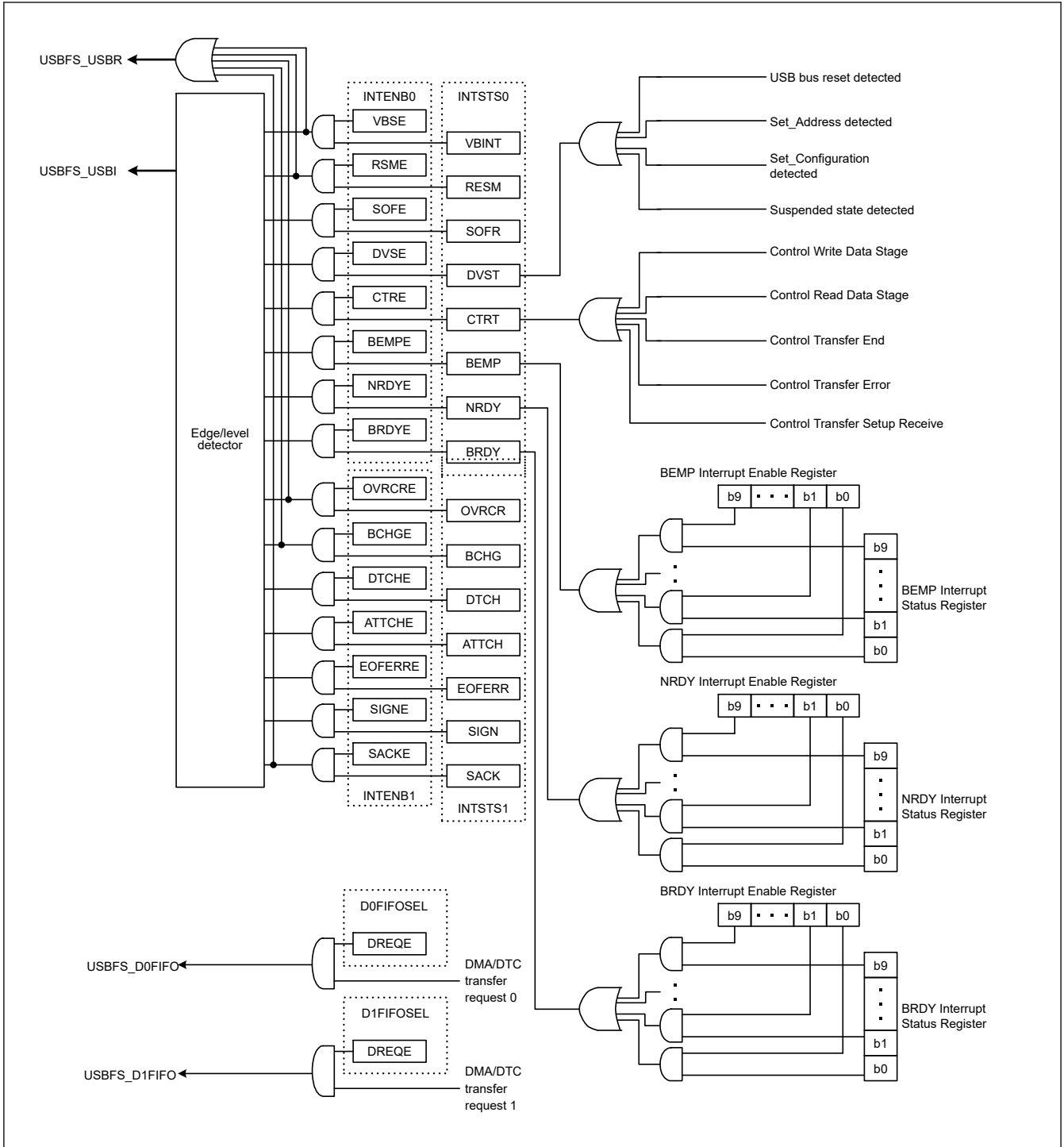


Figure 29.10 USBFS interrupt-related circuits

Table 29.17 shows the interrupts generated by the USBFS.

**Table 29.17 USBFS interrupts**

Interrupt name	Interrupt status flag	DTC activation	DMAC activation	Priority
USBFS_D0FIFO	DMA transfer request 0	Possible	Possible	High
USBFS_D1FIFO	DMA transfer request 1	Possible	Possible	↑
USBFS_USBI	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, overcurrent input change interrupt, bus change interrupt, disconnect detection interrupt during full-speed operation, device connect detection interrupt, EOF error detection interrupt, normal setup operation interrupt, and setup error interrupt	Not possible	Not possible	Low
USBFS_USBR*1	VBUS interrupt, resume interrupt, overcurrent input change interrupt and setup error interrupt	Not possible	Not possible	—

Note 1. Software Standby mode can be canceled. And Deep Software Standby mode 1 also can be canceled, but only dedicated pin (OVRCURA-DS, OVRCURB-DS) can be used for overcurrent pins.

### 29.3.3 Interrupt Descriptions

#### 29.3.3.1 BRDY interrupt

The BRDY interrupt is generated in both host and device controller modes. This section describes the conditions in which the USBFS sets the associated bit in BRDYSTS to 1. Under these conditions, the USBFS generates a BRDY interrupt if the software has set the bit in BRDYENB associated with the given pipe to 1 and the INTENB0.BRDYE bit to 1.

The conditions for generating and clearing the BRDY interrupt depend on the SOFCFG.BRDYM and PIPECFG.BFRE settings for each pipe as follows:

##### (1) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, the USBFS generates an internal BRDY interrupt request trigger and sets the BRDYSTS.PIPEnBRDY bit associated with the selected pipe to 1.

##### For transmitting pipes

- When the DIR bit is changed from 0 to 1 by software
- When packet transmission is complete for a pipe while write-access from the CPU to the FIFO buffer for the pipe is disabled (when the BSTS bit is read as 0)
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is complete
- When the hardware flushes the buffer of the pipe for isochronous transfers
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to transition from the write-disabled to write-enabled state

No request trigger is generated for the DCP, that is, during data transmission for control transfers.

##### For receiving pipes

- When packet reception is successfully complete, enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the given pipe is disabled (when the BSTS bit is read as 0). No request trigger is generated for transactions in which a DATA-PID mismatch has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode. No request trigger is generated until completion of reading data from the currently-read FIFO buffer, even if reception by the other FIFO buffer is complete.

In device controller mode, the BRDY interrupt is not generated in the status stage of control transfers. The PIPEBRDY interrupt status of the selected pipe can be set to 0 by writing 0 to the associated PIPEnBRDY bit through software. In this case, the other PIPEBRDY bit should be set to 1.

Clear the BRDY status before accessing the FIFO buffer.

### (2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1

With these settings, the USBFS generates a BRDY interrupt on completion of reading all data for a single transfer using the receiving pipe, and sets the bit in BRDYSTS associated with the pipe to 1.

On any of the following conditions, the USBFS determines that the last data for a single transfer was received.

- When a short packet including a zero-length packet is received
- When the PIPEn transaction counter register (PIPEnTRN) is used and the number of packets specified in the PIPEnTRN.TRNCNT[15:0] bits are completely received

When the data is completely read after any of these conditions is satisfied, the USBFS determines that all data for a single transfer is completely read.

When a zero-length packet is received while the FIFO buffer is empty, the USBFS determines that all data for a single transfer is completely read when the FRDY bit in the FIFO port control register is 1 and the DTLN[8:0] bits are 0. In this case, to start the next transfer, write 1 to the BCLR bit in the associated port control register through the software. With these settings, the USBFS does not detect a BRDY interrupt for the transmitting pipe.

The PIPEBRDY interrupt status of a pipe can be set to 0 by writing 0 to the associated BRDYSTS.PIPEnBRDY bit through the software. In this case, 1 must be written to the PIPEBRDY bits for the other pipes.

In this mode, do not change the PIPECFG.BFRE bit setting until all data for a single transfer is processed. When it is necessary to change the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pipe must be cleared using the PIPEnCTR.ACLRM bit.

### (3) When SOFCFG.BRDYM = 1 and PIPECFG.BFRE = 0

With these settings, the BRDYSTS.PIPEnBRDY values are linked to the BSTS bit setting for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) are set to 1 or 0 by the USB depending on the FIFO buffer status.

#### For transmitting pipes

The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready. The BRDY interrupt is not generated for the DCP in the transmitting direction even when it is ready for write access.

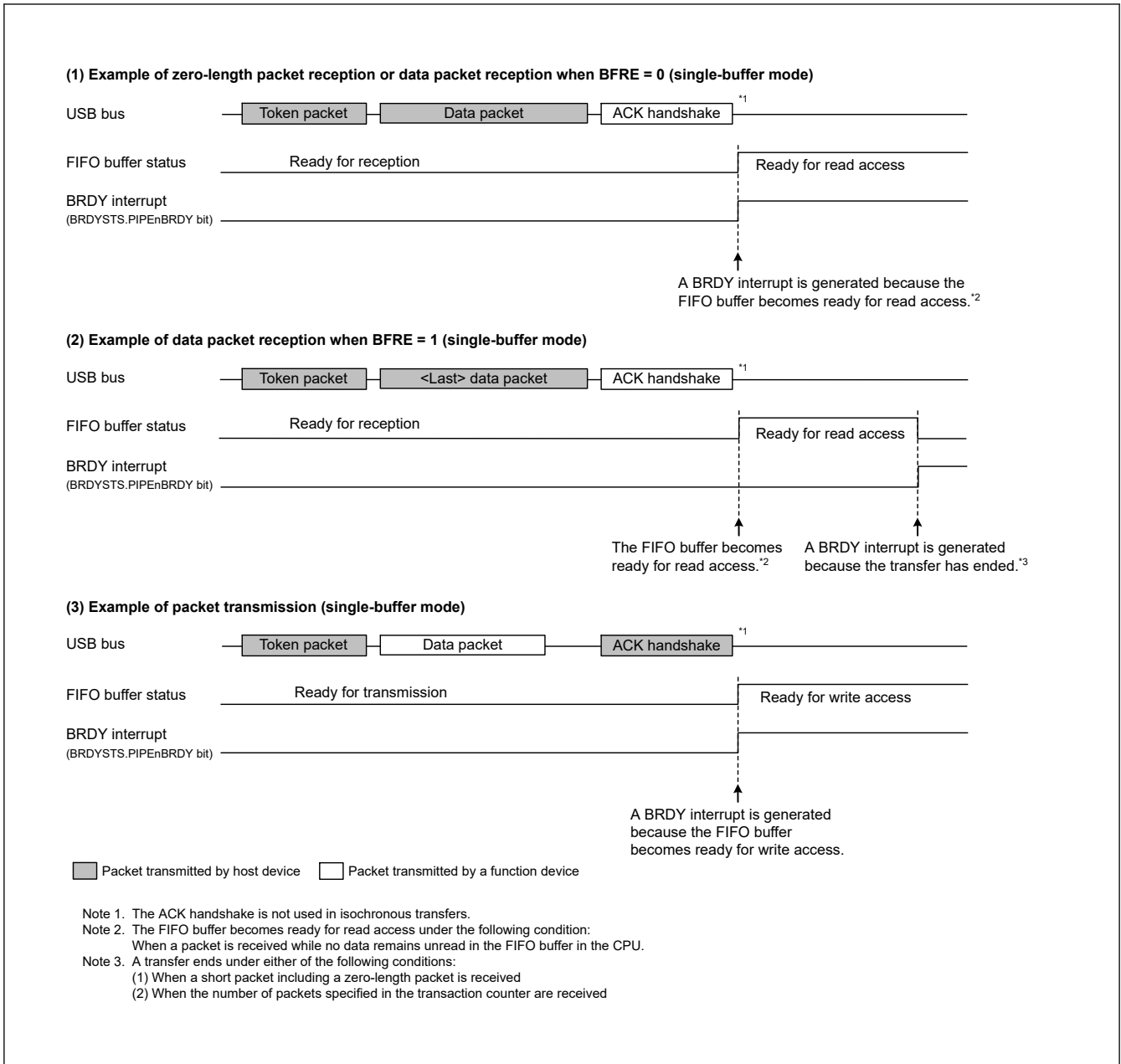
#### For receiving pipes

The BRDY interrupt status bits set to 1 when the FIFO buffer is ready for read access, and set to 0 when all data is read (not ready for read access).

When a zero-length packet is received while the FIFO buffer is empty, the associated bit is set to 1 and the BRDY interrupt is continuously generated until the software writes 1 to BCLR. With this setting, the PIPEnBRDY bit cannot be set to 0 by software.

When the SOFCFG.BRDYM bit is set to 1, set the PIPECFG.BFRE bit for all pipes to 0.

Figure 29.11 shows the timing of BRDY interrupt generation.



**Figure 29.11 Timing of BRDY interrupt generation**

The condition for clearing the INTSTS0.BRDY bit depends on the SOFCFG.BRDYM bit setting, as shown in [Table 29.18](#).

**Table 29.18 Conditions for clearing the BRDY bit**

BRDYM bit	Condition for clearing BRDY bit
0	The USBFS clears the BRDY bit to 0 when all bits in BRDYSTS are set to 0 by software.
1	The USBFS clears the BRDY bit to 0 when the BSTS bits for all pipes have cleared to 0.

### 29.3.3.2 NRDY interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, the USBFS sets the associated PIPEnNRDY bit in NRDYSTS to 1. If the associated bit in NRDYENB is set to 1 by software, the USBFS sets the INTSTS0.NRDY bit to 1 and generates a USBFS interrupt.

This section describes the conditions in which the USBFS generates the internal NRDY interrupt request for a given pipe. The internal NRDY interrupt request is not generated during setup transaction execution in host controller mode. During setup transactions in host controller mode, the SACK or SIGN interrupt is detected.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer in device controller mode.

### (1) In host controller mode

#### For transmitting pipes

On any of the following conditions, the USBFS detects an NRDY interrupt:

- For isochronous transfer pipes, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer. In this case, the USBFS transmits a zero-length packet following the OUT token and sets the associated NRDYSTS.PIPE<sub>N</sub>NRDY bit and the FRMNUM.OVRN bit to 1.
- During communications other than setup transactions on pipes not used for isochronous transfers, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device (when timeout is detected before detection of the handshake packet from the peripheral device)
  - An error is detected in the packet from the peripheral device. In this case, the USBFS sets the associated PIPE<sub>N</sub>NRDY bit to 1 and changes the associated PID[1:0] setting for the pipe to NAK
- During communications other than setup transactions, when the STALL handshake is received from the peripheral device. In this case, the USBFS sets the associated PIPE<sub>N</sub>NRDY bit to 1 and changes the PID[1:0] setting for the associated pipe to STALL (11b).

#### For receiving pipes

- For isochronous transfer pipes, when the time to issue an IN token comes but there is no space available in the FIFO buffer. In this case, the USBFS discards the received data for the IN token and sets the PIPE<sub>N</sub>NRDY bit associated with the pipe and the OVRN bit to 1. When a packet error is detected in the received data for the IN token, the USBFS also sets the FRMNUM.CRCE bit to 1.
- For non-isochronous transfer pipes, when any combination of the following two cases occur three consecutive times:
  - No response is returned from the peripheral device for the IN token issued by the USBFS (when timeout is detected before detection of the DATA packet from the peripheral device)
  - An error is detected in the packet from the peripheral device. In this case, the USBFS sets the associated PIPE<sub>N</sub>NRDY bit to 1 and changes the associated PID[1:0] setting for the pipe to NAK
- For isochronous transfer pipes, when no response is returned from the peripheral device for the IN token (when timeout is detected before detection of the DATA packet from the peripheral device) or an error is detected in the packet from the peripheral device. In this case, the USBFS sets the PIPE<sub>N</sub>NRDY bit associated with the pipe to 1. The PID[1:0] setting for the pipe is not changed.
- For isochronous transfer pipes, when a CRC error or a bit stuffing error is detected in the received data packet. In this case, the USBFS sets the PIPE<sub>N</sub>NRDY bit associated with the pipe and the CRCE bit to 1.
- When the STALL handshake is received. In this case, the USBFS sets the PIPE<sub>N</sub>NRDY bit associated with the pipe to 1 and changes the PID[1:0] setting for the associated pipe to STALL.

### (2) In device controller mode

#### For transmitting pipes

- When an IN token is received while there is no data to be transmitted in the FIFO buffer. In this case, the USBFS generates a NRDY interrupt request on reception of the IN token and sets the NRDYSTS.PIPE<sub>N</sub>NRDY bit to 1. For an isochronous transfer pipe in which an interrupt is generated, the USBFS transmits a zero-length packet and sets the FRMNUM.OVRN bit to 1.

#### For receiving pipes

- When an OUT token is received but there is no space available in the FIFO buffer. For an isochronous transfer pipe in which an interrupt is generated, the USBFS generates a NRDY interrupt request on reception of the OUT token and sets the PIPE<sub>N</sub>NRDY bit to 1 and OVRN bit to 1. For a non-isochronous transfer pipe in which an interrupt is generated, the USBFS generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the PIPE<sub>N</sub>NRDY bit to 1. The NRDY interrupt request is not generated during retransmission



because of a DATA-PID mismatch. In addition, the NRDY interrupt request is not generated if an error occurs in the DATA packet.

- For isochronous transfer pipes, when a token is not received successfully within an interval frame. In this case, the USBFS generates an NRDY interrupt request when the SOF is received, and sets the PIPEnNRDY bit to 1.

Figure 29.12 shows the timing of NRDY interrupt generation in device controller mode.

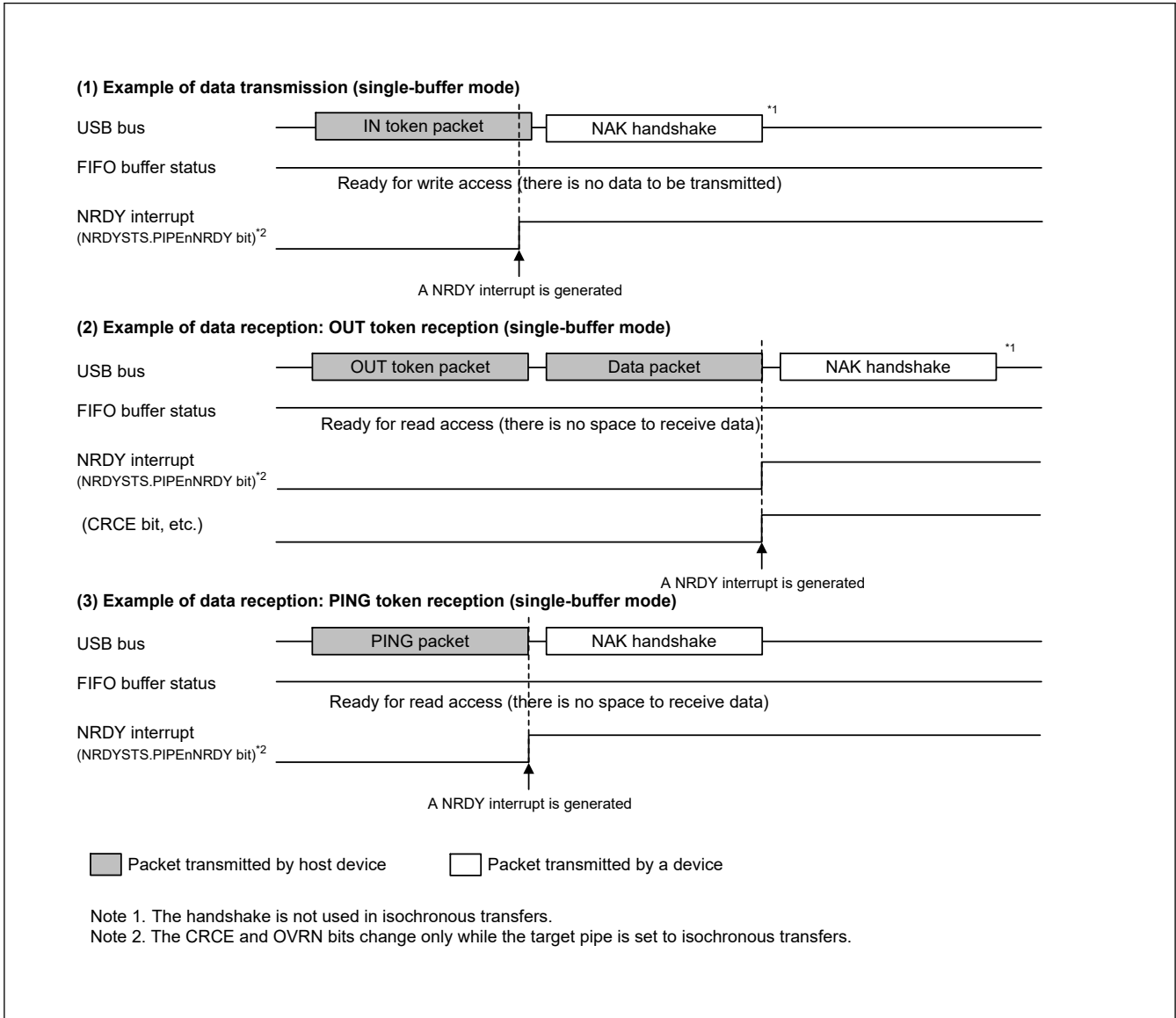


Figure 29.12 Timing of NRDY interrupt generation in device controller mode

### 29.3.3.3 BEMP interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by software, the USBFS sets the associated BEMPSTS.PIPEnBEMP bit to 1. If the associated bit in BEMPENB is set to 1 by software, the USBFS sets the INTSTS0.BEMP bit to 1 and generates a USBFS interrupt. This section describes the conditions in which the USBFS generates an internal BEMP interrupt request.

#### (1) For transmitting pipes

When the FIFO buffer of the associated pipe is empty on completion of transmission, including zero-length packet transmission, and in single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for a non-DCP pipe. The internal BEMP interrupt request is not generated in any of the following conditions:

- When the CPU or DMA/DTC has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode
- When the buffer is cleared (emptied) by setting the PIPEnCTR.ACLRM or the BCLR bit to 1 in the port control register
- When an IN transfer (zero-length packet transmission) is performed during the control transfer status stage in device controller mode

(2) For receiving pipes

When a successfully-received data packet size exceeds the specified maximum packet size. In this case, the USBFS generates a BEMP interrupt request, sets the associated BEMPSTS.PIPEnBEMP bit to 1, discards the received data, and changes the associated PID[1:0] setting for the pipe to STALL (11b). The USBFS returns no response in host controller mode, and returns STALL response in device controller mode.

The internal BEMP interrupt request is not generated in any of the following conditions:

- When a CRC error or a bit stuffing error is detected in the received data
- When a setup transaction is being performed:
  - Writing 0 to the BEMPSTS.PIPEnBEMP bit clears the status
  - Writing 1 to the BEMPSTS.PIPEnBEMP bit has no effect

Figure 29.13 shows the timing of BEMP interrupt generation in device controller mode.

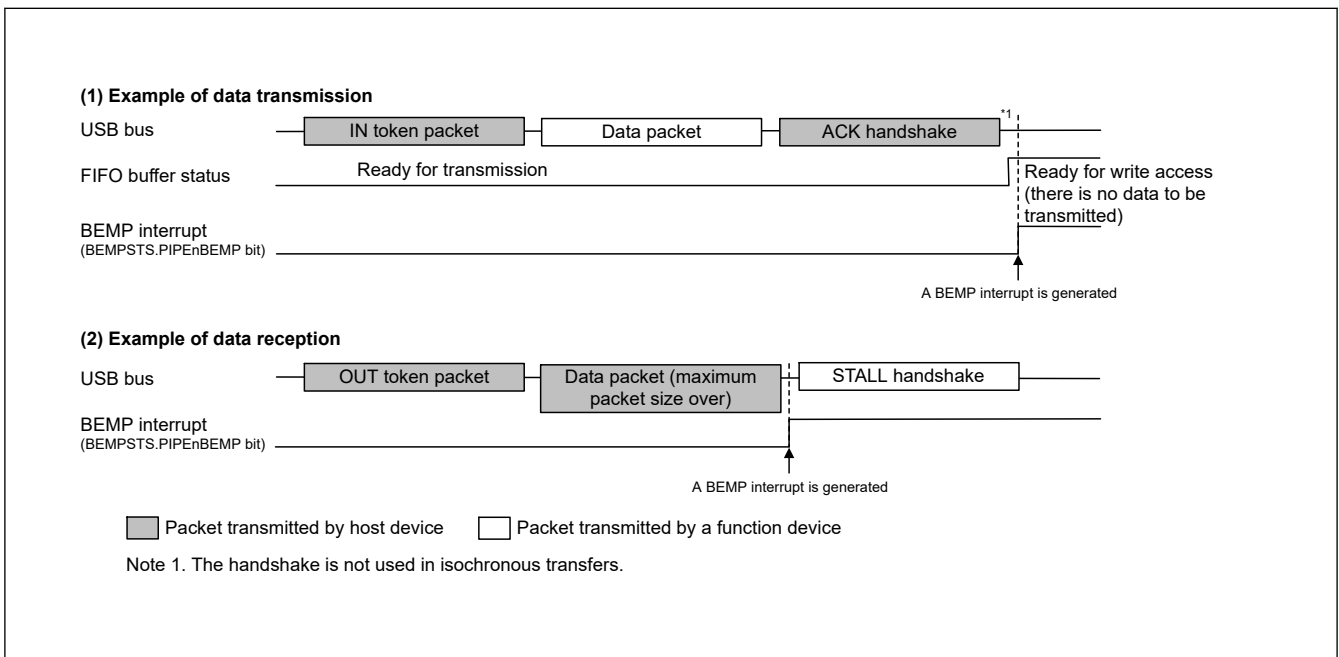


Figure 29.13 Timing of BEMP interrupt generation in device controller mode

29.3.3.4 Device state transition interrupt (device controller mode)

Figure 29.14 shows a diagram of the USBFS device state transitions. The USBFS controls device states and generates device state transition interrupts. However, recovery from the Suspend state (resume signal detection) is detected by means of the resume interrupt. Device state transition interrupts can be enabled or disabled independently in INTENB0. Devices whose states have changed can be checked in the INTSTS0.DVSQ[2:0] bits.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

The USBFS controls device states, and device state transition interrupts can be generated, only in device controller mode.

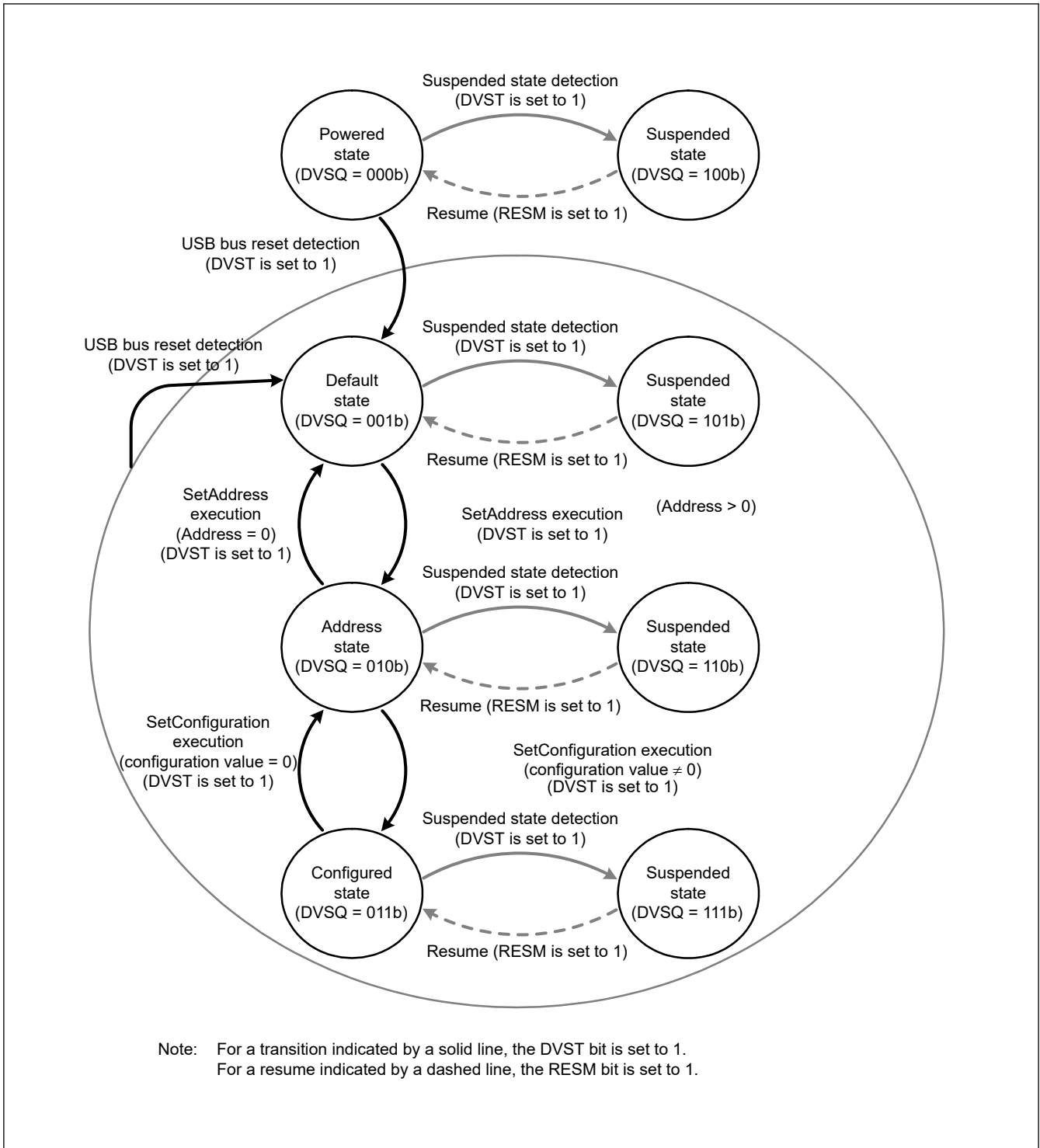


Figure 29.14 Device state transitions

### 29.3.3.5 Control transfer stage transition interrupt (device controller mode)

Figure 29.15 shows a diagram of the control transfer stage transitions of the USBFS. The USBFS controls the control transfer sequence and generates control transfer stage transition interrupts. Control transfer stage transition interrupts can be enabled or disabled independently in INTENB0. Transfer stages that have transitioned can be checked in the INTSTS0.CTSQ[2:0] bits.

Control transfer stage transition interrupts are generated only in device controller mode. This section describes control transfer sequence errors. If an error occurs, the DCPCTR.PID[1:0] bits are set to 1xb (STALL response).

(1) Control read transfer errors

- An OUT token is received but no data is transferred in response to the IN token at the data stage
- An IN token is received at the status stage
- A data packet with DATAPID = DATA0 is received at the status stage

(2) Control write transfer errors

- An IN token is received but no ACK is returned in response to the OUT token at the data stage
- A data packet with DATAPID = DATA0 is received as the first data packet at the data stage
- An OUT token is received at the status stage

(3) Control write no data transfer errors

- An OUT token is received at the status stage

At the control write transfer data stage, if the receive data length exceeds the wLength value of the USB request, it is not recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (INTSTS0.CTRT = 1), the CTSQ[2:0] = 110b value is saved until the CTRT bit is set to 0, clearing the interrupt status. While CTSQ[2:0] = 110b is being saved, no CTRT interrupt for ending the setup stage is generated, even if a new USB request is received. The USBFS saves the setup stage completion status, and it generates a CTRT interrupt after the interrupt status is cleared by software.

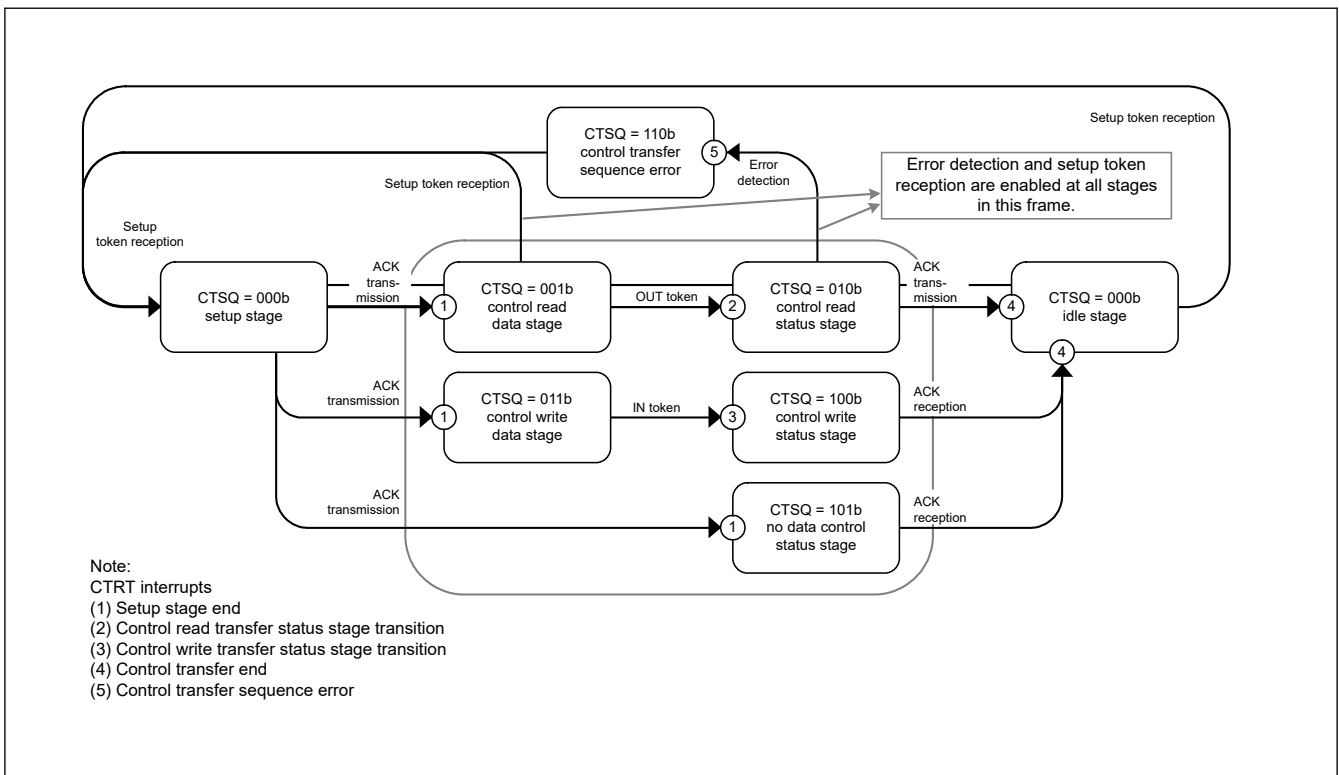


Figure 29.15 Control transfer stage transitions

29.3.3.6 Frame update interrupt

In host controller mode, an interrupt is generated when the frame number is updated.

In device controller mode, an SOFR interrupt is generated when the frame number is updated. The USBFS updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

### 29.3.3.7 VBUS interrupt

When the USB\_VBUS pin level changes, a VBUS interrupt is generated. The level of the USB\_VBUS pin can be checked with the INTSTS0.VBSTS bit. Whether the host controller is connected or disconnected can be confirmed using the VBUS interrupt. If the system is activated with the host controller connected, the first VBUS interrupt is not generated, because there is no change in the USB\_VBUS pin level.

### 29.3.3.8 Resume interrupt

In device controller mode, a resume interrupt is generated when the device state is the Suspend state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the Suspend state is detected by means of the resume interrupt.

In host controller mode, no resume interrupt is generated. Use the BCHG interrupt to detect a change in the USB bus state.

### 29.3.3.9 OVRCCR interrupt

An OVRCCR interrupt is generated when the USB\_OVRCURA, USB\_OVRCURA-DS, USB\_OVRCURB or USB\_OVRCURB-DS pin level has changed. The levels of the USB\_OVRCURA, USB\_OVRCURA-DS, USB\_OVRCURB and USB\_OVRCURB-DS pins can be checked in the SYSSTS0.OVCMON[1:0] flags. The external power supply IC can check whether overcurrent is detected using the OVRCCR interrupt.

For OTG connections, the OVRCCR interrupt allows you to check whether a change is detected in the VBUS comparator.

### 29.3.3.10 BCHG interrupt

A BCHG interrupt is generated when the USB bus state has changed. The BCHG interrupt can be used to detect whether a peripheral device is connected and can also be used to detect a remote wakeup in host controller mode. The BCHG interrupt is generated in both host and device controller modes.

### 29.3.3.11 DTCH interrupt

A DTCH interrupt occurs when a USB bus disconnect is detected in host controller mode. The USBFS detects bus disconnects in compliance with the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software. The pipes enter the wait state for a bus connection to the port, waiting for an ATTCH interrupt to occur. Regardless of the value set in the associated interrupt enable bit, the USBFS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the DTCH interrupt is detected to 0
- Puts the port in which the DTCH interrupt occurred into the idle state

### 29.3.3.12 SACK interrupt

A SACK interrupt is generated when an ACK response for the transmitted setup packet is received from the peripheral device in host controller mode. The SACK interrupt can be used to confirm that the setup transaction is successfully complete.

### 29.3.3.13 SIGN interrupt

A SIGN interrupt is generated when an ACK response for the transmitted setup packet is not correctly received from the peripheral device three consecutive times in host controller mode. The SIGN interrupt can be used to detect no ACK response transmitted from the peripheral device or corruption of an ACK packet.

### 29.3.3.14 ATTCH interrupt

An ATTCH interrupt is generated when J-state or K-state of the full-speed signal level is detected on the USB port for 2.5  $\mu$ s in host controller mode. To be more specific, an ATTCH interrupt is detected on any of the following conditions:

- When K-state, SE0, or SE1 changes to J-state, and J-state continues 2.5  $\mu$ s
- When J-state, SE0, or SE1 changes to K-state, and K-state continues 2.5  $\mu$ s

### 29.3.3.15 EOFERR interrupt

An EOFERR interrupt occurs when the USBFS detects that communication is not complete at the EOF2 timing defined in the USB 2.0 specification.

On interrupt detection, all pipes in which communications are being carried out for the relevant port must be terminated by software, and the port must be re-enumerated. Regardless of the value set in the associated interrupt enable bit, the USBFS hardware:

- Sets the DVSTCTR0.UACT bit for the port in which the EOFERR interrupt is detected to 0
- Puts the port in which the EOFERR interrupt is generated into the idle state

### 29.3.4 Pipe Control

[Table 29.19](#) lists the pipe settings for the USBFS. USB data transfer is performed through logical pipes that the software associates with endpoints. The USBFS provides 10 pipes that are used for data transfer. Set up the pipes based on your system specifications.

**Table 29.19 Pipe settings**

Register name	Bit name	Setting	Notes
DCPCFG PIPECFG	TYPE	Transfer type	Pipes 1 to 9: Settable
	BFRE	BRDY interrupt mode	Pipes 1 to 5: Settable
	DBLB	Double buffer select	Pipes 1 to 5: Settable
	DIR	Transfer direction select	IN or OUT settable
	EPNUM	Endpoint number	Pipes 1 to 9: Settable A value other than 0000b must be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	Pipes 1 and 2: Settable only for bulk transfers Pipes 3 to 5: Settable
DCPMAXP PIPEMAXP	DEVSEL	Device select	Referenced only in host controller mode.
	MXPS	Maximum packet size	Compliant with the USB 2.0 specification.
PIPEPERI	IFIS	Buffer flush	Pipes 1 and 2: Settable only for isochronous transfers Pipes 3 to 9: Setting disabled
	IITV	Interval counter	Pipes 1 and 2: Settable only for isochronous transfers Pipes 3 to 5: Setting disabled Pipes 6 to 9: Settable only in host controller mode
DCPCTR PIPE <sub>n</sub> CTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for pipes 1 to 5.
	SUREQ	Setup request	Settable only for the DCP and controlled in host controller mode
	SUREQCLR	SUREQ clear	Settable only for the DCP and controlled in host controller mode
	ATREPM	Auto response mode	Pipes 1 to 5: Settable only in device controller mode
	ACLRM	Auto buffer clear	Pipes 1 to 9: Settable
	SQCLR	Sequence clear	Clears the data toggle bit
	SQSET	Sequence set	Sets the data toggle bit
	SQMON	Sequence monitor	Monitors the data toggle bit
	PBUSY	Pipe busy status	—
	PID	Response PID	See <a href="#">section 29.3.4.6. Response PID</a> .
PIPE <sub>n</sub> TRE	TRENB	Transaction counter enable	Pipes 1 to 5: Settable
	TRCLR	Current transaction counter clear	Pipes 1 to 5: Settable
PIPE <sub>n</sub> TRN	TRCNT	Transaction counter	Pipes 1 to 5: Settable

### 29.3.4.1 Pipe control register switching procedures

The following bits in the pipe control registers can be changed only when USB communication is prohibited (PID = NAK).

Do not change the following registers and bits when USB communication is enabled (PID = BUF):

- Bits in DCPCFG and DCPMAXP
- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEMAXP, and PIPEPERI
- ATREPM, ACLRM, SQCLR, and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN

To set these bits when USB communication is enabled (PID = BUF):

1. A request to change the bits in the pipe control register occurs.
2. Set the PID[1:0] bits associated with the pipe to NAK.
3. Wait until the associated PBUSY bit clears to 0.
4. Set the bits in the pipe control register.

The following bits in the pipe control registers can be changed only when the selected pipe information is not set in the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Do not set the following registers when the CURPIPE[3:0] bits are set:

- Bits in DCPCFG and DCPMAXP
- Bits in PIPECFG, PIPEMAXP and PIPEPERI

To change pipe information, you must set the CURPIPE[3:0] bits in the port select registers to a pipe other than the one to be changed. For the DCP, the buffer must be cleared using the BCLR bit in the Port Control Register after the pipe information is changed.

### 29.3.4.2 Transfer types

The PIPECFG.TYPE[1:0] bits specify the following transfer types for each pipe:

- DCP: No setting is necessary (fixed at control transfer)
- Pipes 1 and 2: Set to bulk or isochronous transfer
- Pipes 3 to 5: Set to bulk transfer
- Pipes 6 to 9: Set to interrupt transfer

### 29.3.4.3 Endpoint number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to 15.

- DCP: No setting is necessary (fixed at endpoint 0)
- Pipes 1 to 9: Select and set the endpoint numbers from 1 to 15 so that the combination of the PIPECFG.DIR and EPNUM[3:0] bits is unique

### 29.3.4.4 Maximum packet size setting

Specify the maximum packet size for each pipe in the DCPMAXP.MXPS[6:0] and PIPEMAXP.MXPS[9:0] bits. The DCP and pipes 1 to 5 can be set to any of the maximum pipe sizes defined in the USB 2.0 specification. For pipes 6 to 9, the maximum packet size is 64 bytes. Set the maximum packet size as follows before starting a transfer (PID = BUF):

- DCP: Set to 8, 16, 32, or 64
- Pipes 1 to 5: Set to 8, 16, 32, or 64 for bulk transfers
- Pipes 1 and 2: Set between 1 and 256 for isochronous transfers
- Pipes 6 to 9: Set between 1 and 64

### 29.3.4.5 Transaction counter for pipes 1 to 5 in the receiving direction

When the specified number of transactions is complete in the data packet receiving direction, the USBFS recognizes that the transfer ended. Two transaction counters are provided: one is the PIPEnTRN register, which specifies the number of transactions to be executed, and the other is the current counter, which internally counts the number of executed transactions. If the PIPECFG.SHTNAK bit is set to 1, when the current counter value matches the specified number of transactions, the associated PIPEnCTR.PID[1:0] bits are set to NAK and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The data read from PIPEnTRN differs depending on the PIPEnTRE.TRENB setting as follows:

- The TRENB bit = 0: Specified transaction counter value can be read
- The TRENB bit = 1: Current counter value indicating the internally counted number of executed transactions can be read

The following constraints apply when working with the TRCLR bit:

- If the transactions are being counted and PID = BUF, the current counter cannot be cleared
- If there is any data left in the buffer, the current counter cannot be cleared

### 29.3.4.6 Response PID

Specify the response PID for each pipe in the PID[1:0] bits in DCPCTR and PIPEnCTR. This section describes the USBFS operation with different response PID settings.

#### (1) Software response PID settings in host controller mode

Select the response PID to specify the execution of transactions as follows:

- NAK setting: Using pipes is disabled and no transactions are executed
- BUF setting: Transactions are executed based on the FIFO buffer state:
  - OUT direction: An OUT token is issued if the FIFO buffer contains transmit data.
  - IN direction: An IN token is issued if the FIFO buffer is not full and can receive data.
- STALL setting: Using pipes is disabled and no transactions are executed

Note: Use the DCPCTR.SUREQ bit to execute setup transactions for the DCP.

#### (2) Software response PID settings in device controller mode

Select the response PID to respond as follows to transactions from the host:

- NAK setting: A NAK response is returned to all generated transactions
- BUF setting: A response is returned to transactions based on the FIFO buffer
- STALL setting: A STALL response is returned to all generated transactions

Note: For setup transactions, an ACK response is always returned, regardless of the PID[1:0] bits setting, and the USB request is stored in the register.

Sections (3) and (4) describe situations in which the USBFS writes to the PID[1:0] bits because of specific transaction results.

#### (3) Hardware response PID settings in host controller mode

- NAK setting: PID = NAK is set in the following cases, and issuing of tokens is automatically stopped:
  - When a non-isochronous transfer is performed and an NRDY interrupt is generated  
(For details, see [section 29.3.3.2. NRDY interrupt](#).)
  - If a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
  - If transaction counting ends when the SHTNAK bit is set to 1 for bulk transfers
- BUF setting: The USBFS does not write this setting.



- STALL setting: PID = STALL is set in the following cases, and issuing of tokens is automatically stopped:
  - When STALL is received in response to a transmitted token
  - When a received data packet exceeds the maximum packet size

#### (4) Hardware response PID settings in device controller mode

- NAK setting: PID = NAK is set in the following cases, and a NAK response is returned to transactions:
  - When the setup token is received normally (DCP only)
  - If transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit is set to 1 for bulk transfers
- BUF setting: There is no BUF writing by the USBFS.
- STALL setting: PID = STALL is set in the following cases, and a STALL response is returned to transactions:
  - When a received data packet exceeds the maximum packet size
  - When a control transfer sequence error is detected (DCP only)

#### 29.3.4.7 Data PID sequence bit

The USBFS automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit toggles on ACK handshake reception. When data is received, the sequence bit toggles on ACK handshake transmission. The SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

In device controller mode when control transfers are used, the USBFS automatically sets the sequence bit for stage transitions. DATA1 is returned when the setup stage ends. The sequence bit is not referenced and PID = DATA1 is returned in the status stage. Therefore, no software settings are required. However, in host controller mode when control transfers are used, the sequence bit must be set by software for the stage transitions.

For ClearFeature requests for transmission or reception, the data PID sequence bit must be set by software in both host and device controller modes.

#### 29.3.4.8 Response PID = NAK function

The USBFS provides a function for disabling pipe operation (PID response = NAK) when the final data packet of a transaction is received. The USBFS automatically distinguishes this based on reception of a short packet or the transaction counter. Enable this function by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the FIFO buffer, using this function enables reception of data packets in transfer units. If pipe operation is disabled, the software must enable the pipe again (PID response = BUF).

The response PID = NAK function can be used only for bulk transfers.

#### 29.3.4.9 Auto response mode

For bulk transfer pipes (1 to 5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (PIPECFG.DIR = 0), OUT-NAK mode is invoked, and during an IN transfer (DIR = 1), null auto response mode is invoked.

#### 29.3.4.10 OUT-NAK mode

For bulk OUT transfer pipes, NAK is returned in response to an OUT token, and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To transition from normal mode to OUT-NAK mode, specify OUT-NAK mode while pipe operation is disabled (PID[1:0] = 00b for NAK response). Next enable pipe operation (PID[1:0] = 01b for BUF response), on which OUT-NAK mode becomes valid. If an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To transition from OUT-NAK mode to normal mode, cancel OUT-NAK mode while pipe operation is disabled (NAK). Next enable pipe operation (BUF). In normal mode, reception of OUT data is enabled.

### 29.3.4.11 Null auto response mode

For bulk IN transfer pipes, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To transition from normal mode to null auto response mode, specify null auto response mode while pipe operation is disabled (response PID = NAK). Next enable pipe operation (response PID = BUF) on which null auto response mode becomes valid. Before setting null auto response mode, check that PIPEnCTR.INBUFM = 0, because the mode can be set only when the buffer is empty. If the INBUFM bit is 1, empty the buffer using the PIPEnCTR.ACLRM bit. Do not write data from the FIFO port while a transition to null auto response mode is being made.

To transition from null auto response mode to normal mode, keep pipe operation disabled (response PID = NAK) for the period of the zero-length packet transmission (about 10  $\mu$ s) before canceling the null auto response mode. In normal mode, data can be written from the FIFO port, so packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

### 29.3.5 FIFO Buffer

The USBFS provides a FIFO buffer for data transfers, and it manages the memory area used for each pipe. The FIFO buffer has two states depending on whether the access right is assigned to the system (CPU side) or the USBFS (SIE side).

#### (1) Buffer status

Table 29.20 and Table 29.21 show the buffer status in the USBFS. The FIFO buffer status can be confirmed using the DCPCTR.BSTS and PIPEnCTR.INBUFM bits. The transfer direction for the FIFO buffer can be specified in either the PIPECFG.DIR or CFIFOSEL.ISEL bit (when DCP is selected).

The INBUFM bit is valid for pipes 1 to 5 in the transmitting direction.

When a transmitting pipe uses double buffering, the software can read the BSTS bit to monitor the FIFO buffer status on the CPU side and the INBUFM bit to monitor the FIFO buffer status on the SIE side. When write access to the FIFO port by the CPU or DMA/DTC is slow and the buffer empty status cannot be determined using the BEMP interrupt, the software can use the INBUFM bit to confirm the end of transmission.

**Table 29.20 Buffer status indicated in the BSTS bit**

ISEL or DIR	BSTS	FIFO buffer status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet is received. Reading from the FIFO port is allowed. When a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	Transmission has not completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	Transmission is complete. CPU write is allowed.

**Table 29.21 Buffer status indicated in the INBUFM bit**

DIR	INBUFM	FIFO buffer status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	Transmission is complete. There is no data waiting to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

### 29.3.6 FIFO Buffer Clearing

Table 29.22 shows the methods for clearing the FIFO buffer. The FIFO buffer can be cleared using BCLR bit in the port control register, DnFIFOSEL.DCLRM, or the PIPEnCTR.ACLRM bit.

Single or double buffering can be selected for pipes 1 to 5 in the PIPECFG.DBLB bit.

**Table 29.22 Buffer clearing methods**

FIFO buffer clearing mode	Clearing FIFO buffer on the CPU side	Mode for automatically clearing the FIFO buffer after reading the specified pipe data	Auto buffer clear mode for discarding all received packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

### (1) Auto buffer clear mode function

The USBFS discards all received data packets if the PIPEnCTR.ACLRm bit is set to 1. If a correct data packet is received, the ACK response is returned to the host controller. The auto buffer clear mode function can only be set in the FIFO buffer reading direction.

Setting the ACLRM bit to 1 and then to 0 clears the FIFO buffer of the selected pipe regardless of the access direction. An access cycle of at least 100 ns is required for the internal hardware sequence processing between ACLRM = 1 and ACLRM = 0.

## 29.3.7 FIFO Port Functions

Table 29.23 shows the settings for the FIFO port functions. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, set the BVAL flag in the port control register to end writing. To send a zero-length packet, use the BCLR bit to clear the buffer, and then set the BVAL flag to end writing.

In reading, reception of new packets is automatically enabled when all data is read. Data cannot be read when a zero-length packet is received (DTLN[8:0] = 0), so the buffer must be cleared with the BCLR bit. The length of the receive data can be confirmed in the DTLN[8:0] bits in the port control register.

**Table 29.23 FIFO port function settings**

Register name	Bit name	Description
CFIFOSEL, DnFIFOSEL (n = 0, 1)	RCNT	Selects DTLN[11:0] read mode
	REW	FIFO buffer rewind (re-read, rewrite)
	DCLRM	Automatically clears receive data for a specified pipe after the data is read (only for DnFIFO)
	DREQE	Enables DMA/DTC transfers (only for DnFIFO)
	MBW	FIFO port access bit width
	BIGEND	Selects FIFO port endian
	ISEL	FIFO port access direction (only for DCP)
	CURPIPE	Selects the current pipe
CFIFOCTR, DnFIFOCTR (n = 0, 1)	BVAL	Ends writing to the FIFO buffer
	BCLR	Clears the FIFO buffer on the CPU side
	DTLN	Checks the length of receive data

### (1) FIFO port selection

Table 29.24 shows the pipes that can be selected with the different FIFO ports. The pipe to be accessed must be selected in the CURPIPE[3:0] bits in the port select register. After the pipe is selected, the software must check whether the written value can be read correctly from the CURPIPE[3:0] bits. (If the previous pipe number is read, it indicates that the USBFS is modifying the pipe.) Next, the software checks that the FRDY bit in the port control register is 1.

In addition, the software must specify the bus width to be accessed in the MBW bit in the port select register. The FIFO buffer access direction conforms to the PIPECFG.DIR setting. For the DCP only, the ISEL bit in the port select register determines the direction.

**Table 29.24** FIFO port access by pipe

Pipe	Access method	Ports that can be used
DCP	CPU access	CFIFO port register
Pipes 1 to 9	CPU access	<ul style="list-style-type: none"> <li>• CFIFO port register</li> <li>• D0FIFO/D1FIFO port register</li> </ul>
	DMA/DTC access	D0FIFO/D1FIFO port register

## (2) REW bit

It is possible to temporarily stop access to a pipe currently being accessed, access a different pipe, and then continue processing for the first pipe again. The REW bit in the port select register is used for this processing.

If a pipe is selected in the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, the pointer used for reading from and writing to the FIFO buffer is reset, and reading or writing can be carried out from the first byte. If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, the software must check that the FRDY bit in the port control register is 1 after selecting a pipe.

## 29.3.8 DMA Transfers (D0FIFO and D1FIFO Ports)

### (1) Overview of DMA transfers

For pipes 1 to 9, the FIFO port can be accessed using the DMAC. When buffer access for a pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.

Select the unit of transfer to the FIFO port in the DnFIFOSEL.MBW bit, and select the pipe targeted for the DMA transfer in the DnFIFOSEL.CURPIPE[3:0] bits. Do not change the selected pipe during the DMA transfer.

### (2) DnFIFO auto clear mode (D0FIFO and D1FIFO port reading direction)

If 1 is set in the DnFIFOSEL.DCLRM bit, the USBFS automatically clears the FIFO buffer of the selected pipe when reading of data from the FIFO buffer is complete.

Table 29.25 shows the packet reception and FIFO buffer clearing processing by software for each of the settings.

As shown in the table, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.

The DnFIFO auto clear mode can only be set in the FIFO buffer reading direction.

**Table 29.25** Packet reception and FIFO buffer clearing processing by software

Buffer status when packet is received	Register setting			
	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	No clearing required	No clearing required	No clearing required	No clearing required
Zero-length packet reception	Clearing required	Clearing required	No clearing required	No clearing required
Normal short packet reception	No clearing required	Clearing required	No clearing required	No clearing required
Transaction count end	No clearing required	Clearing required	No clearing required	No clearing required

## 29.3.9 Control Transfers Using the DCP

The DCP is used for data transfers in the control transfer data stage. The FIFO buffer of the DCP is a 64-byte single buffer with a fixed area for both control reads and control writes. The FIFO buffer can be accessed only through the CFIFO port.

### 29.3.9.1 Control transfers in host controller mode

#### (1) Setup stage

The USQREQ, USBVAL, USBINDX, and USBLENG registers are used to transmit USB requests for setup transactions. Writing the setup packet data to the registers and then writing 1 to the DCPCTR.SUREQ bit transmits the specified data for the setup transaction. On completion of the transaction, the SUREQ bit clears to 0. Do not change these USB request registers while SUREQ = 1.

When an attached function device is detected, the software must issue the first setup transaction for the device using this sequence with the DCPMAXP.DEVSEL[3:0] bits cleared to 0 and the DEVADD0.USBSPD[1:0] bits set appropriately.

When an attached function device is shifted to the Address state, the software must issue setup transactions using this sequence with the assigned USB address set in the DEVSEL[3:0] bits and the bits in DEVADDn corresponding to the specified USB address set appropriately. For example, when PIPEMAXP.DEVSEL[3:0] = 0010b, make appropriate settings in DEVADD2. When PIPEMAXP.DEVSEL[3:0] = 0101b, make appropriate settings in DEVADD5.

When the setup transaction data is sent, an interrupt request is generated based on the response from the peripheral device (SIGN or SACK bit in INTSTS1). This interrupt request allows the software to check the setup transaction result.

A DATA0 data packet (USB request) for the setup transaction is always transmitted regardless of the status of the DCPCTR.SQMON bit.

#### (2) Data stage

The data stage is used to transfer data using the DCP FIFO buffer.

Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit. Specify the transfer direction in the DCPCFG.DIR bit.

For the first data packet of the data stage, the data PID must be transferred as DATA1. Set data PID = DATA1 in the DCPCTR.SQSET bit and set the PID bits = BUF. Completion of data transfer is detected using the BRDY or BEMP interrupt.

For control write transfers, when the number of data bytes to be sent is an integer multiple of the maximum packet size, the software must send a zero-length packet at the end.

#### (3) Status stage

The status stage is used for zero-length packet data transfers in the reverse direction of the data stage. As in the data stage, data is transferred using the DCP FIFO buffer. Transactions are executed using the same procedure as the data stage.

Data packets in the status stage must be transmitted and received with the data PID set to DATA1 using the DCPCTR.SQSET bit.

When a zero-length packet is received, check the receive-data length in the CFIFOCTR.DTLN[8:0] bits after a BRDY interrupt is generated, and then clear the FIFO buffer using the BCLR bit.

### 29.3.9.2 Control transfers in device controller mode

#### (1) Setup stage

The USBFS sends an ACK response to a normal setup packet for the USBFS. The USBFS operates in the setup stage as follows:

On receiving a new setup packet, the USBFS sets the following bits:

- Sets the INTSTS0.VALID bit to 1
- Sets the DCPCTR.PID[1:0] bits to NAK
- Sets the DCPCTR.CCPL bit to 0

When the USBFS receives a data packet following a setup packet, it stores the USB request parameters in USBREQ, USBVAL, USBINDX, and USBLENG.

Before performing the response processing for a control transfer, set the VALID flag to 0. When the VALID bit = 1, PID = BUF cannot be set, and the data stage cannot be terminated.

Using the VALID bit function, the USBFS can suspend a request being processed when it receives a new USB request during a control transfer and return a response to the latest request.

In addition, the USBFS automatically detects the direction bit (bmRequestType bit [8]) and the request data length (wLength) in the received USB request. It distinguishes between control read transfers, control write transfers, and no-data control transfers, and it controls stage transitions. For an incorrect sequence, a sequence error occurs in the control transfer stage transition interrupt, and the interrupt is reported to the software. For a diagram of the stage control by the USBFS, see [Figure 29.15](#).

## (2) Data stage

The DCP must be used to execute data transfers for received USB requests. Before accessing the DCP FIFO buffer, specify the access direction in the CFIFOSEL.ISEL bit.

If the transfer data is larger than the size of the DCP FIFO buffer, execute the data transfer using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

## (3) Status stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to BUF.

After this setting is made, the USBFS automatically executes the status stage based on the data transfer direction determined at the setup stage. The procedure is as follows:

- For control read transfers  
The USBFS receives a zero-length packet from the USB host and transmits an ACK response.
- For control write transfers and no-data control transfers  
The USBFS transmits a zero-length packet and receives an ACK response from the USB host.

## (4) Control transfer auto response function

The USBFS automatically responds to a correct SET\_ADDRESS request. If any of the following errors occurs in the SET\_ADDRESS request, a response from the software is necessary.

- bmRequestType is not 0x00: Any transfer other than a control write transfer
- wIndex is not 0x00: Request error
- wLength is not 0x00: Any transfer other than a no-data control transfer
- wValue is larger than 0x7F: Request error
- INTSTS0.DVSQ[2:0] are 011b (Configured state): Control transfer of a device state error

For all requests other than the SET\_ADDRESS request, a response is required from the corresponding software.

### 29.3.10 Bulk Transfers (Pipes 1 to 5)

The FIFO buffer usage (single/double buffer setting) is configurable for bulk transfers. The USBFS provides the following functions for bulk transfers:

- BRDY interrupt function (PIPECFG.BFRE bit), see [section 29.3.3.1. BRDY interrupt](#)
- Transaction count function (PIPEnTRE.TRENB, TRCLR, and PIPEnTRN.TRNCNT[15:0] bits), see [section 29.3.4.5. Transaction counter for pipes 1 to 5 in the receiving direction](#)
- Response PID = NAK function (PIPECFG.SHTNAK bit), see [section 29.3.4.8. Response PID = NAK function](#)
- Auto response mode (PIPEnCTR.ATREPM bit), see [section 29.3.4.9. Auto response mode](#)

### 29.3.11 Interrupt Transfers (Pipes 6 to 9)

In device controller mode, the USBFS performs interrupt transfers based on the timing dictated by the host controller.

In host controller mode, the software can set the timing for issuing tokens using the interval counter.

### 29.3.11.1 Interval counter for interrupt transfers in host controller mode

Specify the transaction interval for interrupt transfers in the PIPEPERI.IITV[2:0] bits. The USBFS issues interrupt transfer tokens based on this interval.

#### (1) Counter initialization

The USBFS initializes the interval counter under the following conditions:

- Power-on reset  
This initializes the IITV[2:0] bits.
- FIFO buffer initialization using the PIPEnCTR.ACLRM bit:  
This does not initialize the IITV[2:0] bits, but does initialize the count value. Setting the PIPEnCTR.ACLRM bit to 0 starts counting from the value set in IITV[2:0].

The interval counter is not initialized in the following case:

- USB bus reset or USB suspended  
The IITV[2:0] bits are not initialized. Setting 1 to the DVSTCTR0.UACT bit starts counting from the value saved before entering the USB bus reset state or USB suspend state.

#### (2) Operation when tokens cannot be transmitted or received even on token generation

No token is generated in the following cases even at token generation time. In these cases, the USBFS tries to execute the transaction in the next interval.

- When the PID is set to NAK or STALL
- When the FIFO buffer is full at token transmit time in the receiving (IN) direction
- When there is no data to be transmitted in the FIFO buffer at token transmit time in the transmitting (OUT) direction

### 29.3.12 Isochronous Transfers (Pipes 1 and 2)

The USBFS provides the following functions for isochronous transfers:

- Notification of isochronous transfer error
- Interval counter specified in the PIPEPERI.IITV[2:0] bits
- Isochronous IN transfer data setup control (IDLY function)
- Isochronous IN transfer buffer flush function specified in the PIPEPERI.IFIS bit

#### 29.3.12.1 Error detection in isochronous transfers

The USBFS provides a function for detecting the errors described in this section, so that when errors occur in isochronous transfers, they can be controlled by software. [Table 29.26](#) and [Table 29.27](#) show the priority order for errors detected by the USBFS and the associated interrupts.

##### **PID errors**

- The PID value of the received packet is invalid.

##### **CRC errors and bit stuffing errors**

- A CRC error is found in a received packet or the bit stuffing is illegal.

##### **Maximum packet size exceeded**

- The data size of the received packet exceeds the specified maximum packet size.

##### **Overrun and underrun errors**

In host controller mode:

- The FIFO buffer is full at token transmit time in the IN (receiving) direction
- There is no data to be sent in the FIFO buffer at token transmit time in the OUT (transmitting) direction

In device controller mode:

- There is no data to be sent in the FIFO buffer at token receive time in the IN (transmitting) direction
- The FIFO buffer is full at token receive time in the OUT (receiving) direction

### Interval errors

In device controller mode, the following cases are treated as an interval error:

- Failure to receive an IN token in the interval frame during an isochronous IN transfer
- Failure to receive an OUT token in the interval frame during an isochronous OUT transfer

**Table 29.26 Error detection for token transmission and reception**

Detection priority	Error	Generated interrupt and status
1	PID error	No interrupts are generated in either host or device controller mode (ignored as a corrupted packet)
2	CRC or bit stuffing error	No interrupts are generated in either host or device controller mode (ignored as a corrupted packet)
3	Overflow or underflow error	An NRDY interrupt is generated to set the FRMNUM.OVRN bit to 1 in both host and device controller modes. In device controller mode, a zero-length packet is transmitted in response to an IN token. No data packets are received in response to OUT token.
4	Interval error	An NRDY interrupt is generated in device controller mode. No interrupt is generated in host controller mode.

**Table 29.27 Error detection for data packet reception**

Detection priority	Error	Generated interrupt and status
1	PID error	No interrupts are generated (ignored as a corrupted packet)
2	CRC or bit stuffing error	An NRDY interrupt is generated and the FRMNUM.CRCE bit sets to 1 in both host and device controller modes
3	Maximum packet size exceeded error	A BEMP interrupt is generated and the PID[1:0] bits set to STALL in both host and device controller modes

### 29.3.12.2 DATA-PID

In device controller mode, the USBFS responds to a received PID as follows:

#### (1) IN direction

- DATA0: Transmitted as data packet PID
- DATA1: Not transmitted
- DATA2: Not transmitted
- mData: Not transmitted

#### (2) OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets ignored
- mData: Packets ignored

### 29.3.12.3 Interval counter

The isochronous transfer interval can be set in the PIPEPERI.IITV[2:0] bits. In device controller mode, the interval counter enables the functions as shown in [Table 29.28](#). In host controller mode, the USBFS generates the token issuance timing, and the interval counter operation is the same as that for interrupt transfers.



**Table 29.28 Interval counter functions in device controller mode**

Transfer direction	Function	Conditions for detection
IN	Transmit buffer flush	Failure to receive an IN token successfully in the interval frame during an isochronous IN transfer.
OUT	Notification of no reception of token	Failure to receive an OUT token successfully in the interval frame during an isochronous OUT transfer.

The interval count is performed when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is corrupt. The frame interval can be set to  $2^{IITV}$  frames.

**(1) Counter initialization in device controller mode**

The USBFS initializes the interval counter under the following conditions:

- Power-on reset  
This initializes the PIPEPERI.IITV[2:0] bits.
- FIFO buffer initialization using the ACLRM bit  
This does not initialize the IITV[2:0] bits, but does initialize the count value.

After the interval counter is initialized, the interval count starts under one of the following conditions when a packet is transferred successfully:

- An SOF is received after data is transmitted in response to an IN token when PID = BUF
- An SOF is received after data is received in response to an OUT token when PID = BUF

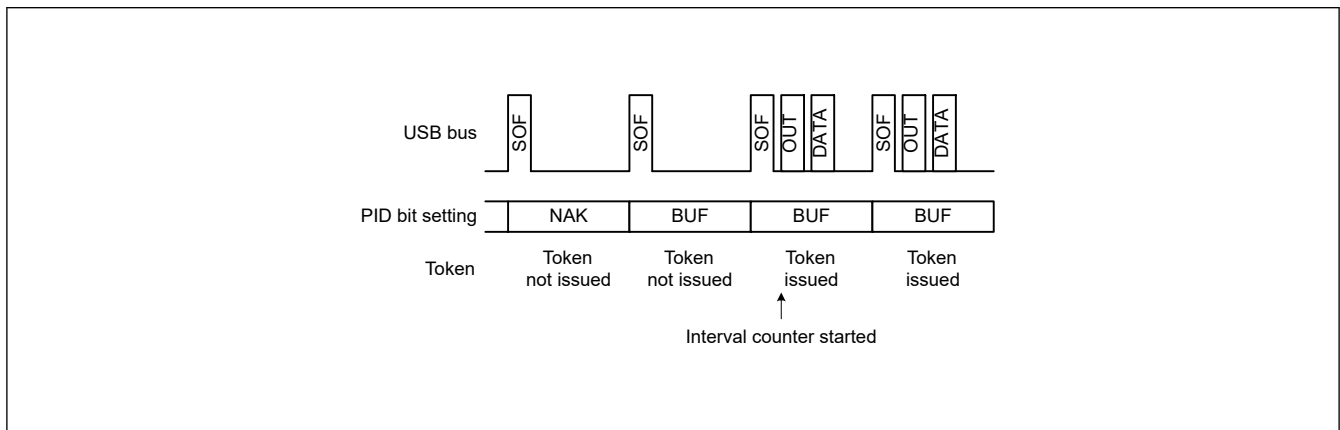
The interval counter is not initialized in the following conditions:

- When the PID[1:0] bits are set to NAK or STALL  
This does not stop the interval timer. The USBFS attempts the transaction in the next interval.
- When the USB bus is reset or USBFS is suspended  
This does not initialize the IITV[2:0] bits. When an SOF is received, the interval counter starts counting from the value set before SOF was received.

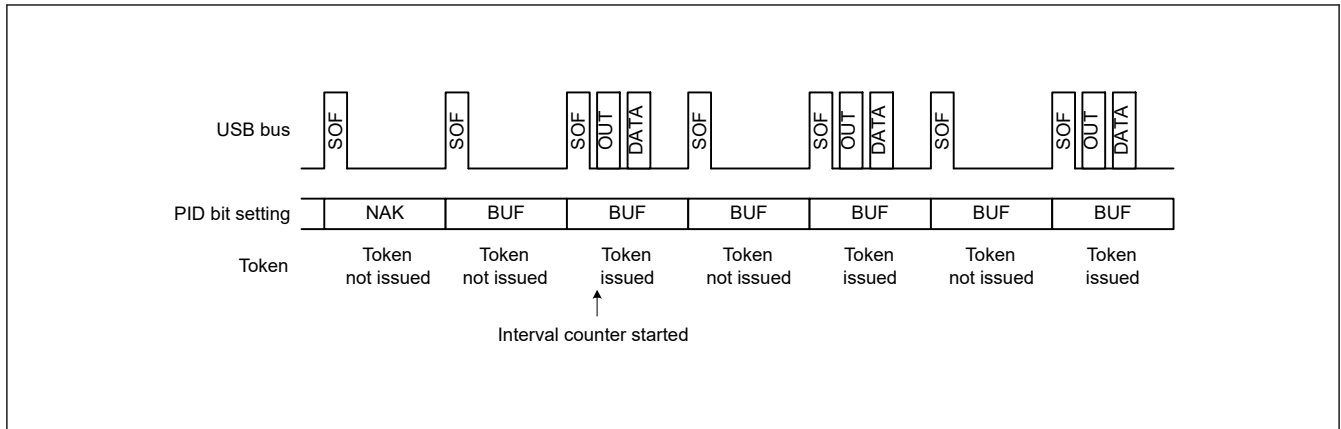
**(2) Interval counting and transfer control in host controller mode**

The USBFS controls the interval between token issuance operations based on the PIPEPERI.IITV[2:0] bits settings. Specifically, the USBFS issues a token for a selected pipe once every  $2^{IITV}$  frames.

The USBFS starts counting the token issuance interval at the frame following the frame in which the PID[1:0] bits are set to BUF by software.



**Figure 29.16 Token issuance when IITV = 0**



**Figure 29.17 Token issuance when IITV = 1**

When the selected pipe is set for isochronous transfers, the USBFS performs the following operation in addition to controlling the token issuance interval. The USBFS issues a token even when the NRDY interrupt generation condition is satisfied.

#### When the selected pipe is for isochronous IN transfers

The USBFS generates an NRDY interrupt when the USBFS issues an IN token but does not successfully receive a packet from a peripheral device (no response or packet error).

The USBFS sets the FRMNUM.OVRN bit to 1, generating an NRDY interrupt, when the time to issue an IN token occurs while the USBFS cannot receive data because the FIFO buffer is full, because the CPU or DMAC/DTC is too slow in reading data from the FIFO buffer.

#### When the selected pipe is for isochronous OUT transfers

The USBFS sets the OVRN bit to 1, generating an NRDY interrupt and transmitting a zero-length packet, when the time to issue an OUT token comes while there is no data to be transmitted in the FIFO buffer, or because the CPU or DMAC/DTC is too slow in writing data to the FIFO buffer.

The token issuance interval is reset on any of the following conditions:

- When the USBFS is reset through a reset pin  
This initializes the IITV[2:0] bits.
- When the PIPEnCTR.ACLRM bit is set to 1 by software

### (3) Interval counting and transfer control in device controller mode

#### When the selected pipe is for isochronous OUT transfers

The USBFS generates an NRDY interrupt when it fails to receive a data packet within the interval set in the PIPEPERI.IITV[2:0] bits.

The USBFS also generates an NRDY interrupt when it fails to receive data because of a CRC error or other errors contained in the data packet or because the FIFO buffer is full.

The NRDY interrupt is generated on SOF packet reception. Even if the SOF packet is corrupted, internal interpolation allows the interrupt to be generated when the SOF packet is received. However, when the IITV bits are set to a value other than 0, the USBFS generates an NRDY interrupt on receiving an SOF packet for every interval after interval counting starts.

When the PID[1:0] bits are set to NAK by software after starting the interval timer, the USBFS does not generate an NRDY interrupt on receiving an SOF packet.

The timing for starting interval counting depend on the IITV[2:0] setting as follows:

- When the IITV[2:0] bits = 0:  
Interval counting starts at the next frame after the software changes the PID[1:0] bits of the selected pipe to BUF.
- When the IITV[2:0] bits  $\neq$  0:  
Interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe are changed to BUF.

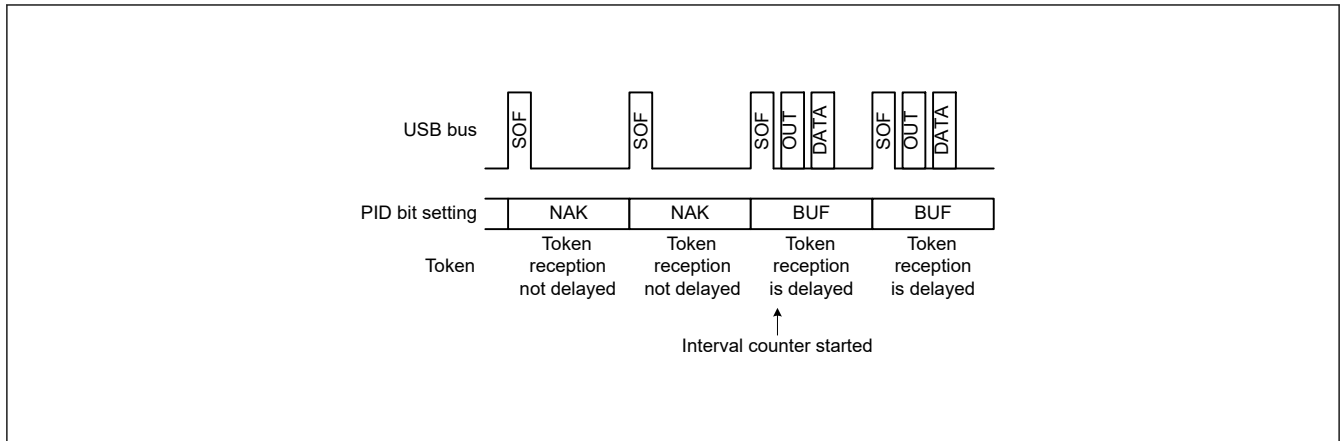


Figure 29.18 Relationship between frames and expected token reception when IITV[2:0] = 0

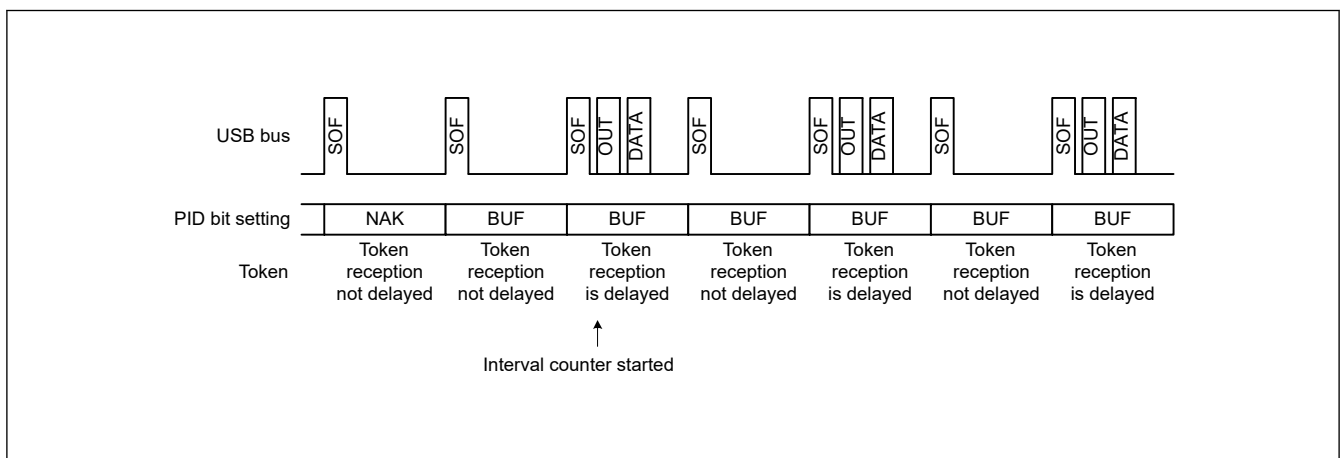


Figure 29.19 Relationship between frames and expected token reception when IITV[2:0] ≠ 0

**When the selected pipe is for isochronous IN transfers**

The PIPEPERI.IFIS bit must be 1 for this use case. When IFIS = 0, the USBFS transmits a data packet in response to a received IN token regardless of the PIPEPERI.IITV[2:0] setting.

When IFIS is 1 and there is data to be transmitted in the FIFO buffer, the USBFS clears the FIFO buffer when it fails to receive an IN token in the frame at the interval set in the IITV[2:0] bits.

The USBFS also clears the FIFO buffer when it fails to receive an IN token successfully because of a bus error, such as a CRC error, contained in the IN token.

The FIFO buffer is cleared on SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared when the SOF packet is received.

The timing to start interval counting depends on the IITV[2:0] setting, as with OUT transfers.

The interval is counted on any of the following conditions in device controller mode:

- When a hardware reset is applied to the USBFS (which also sets the IITV[2:0] bits to 000b)
- When the PIPEnCTR.ACLR bit is set to 1 by software
- When the USBFS detects a USB bus reset

**(4) Transmit data setup for isochronous transfers in device controller mode**

With isochronous data transmission using the USBFS in device controller mode, after data is written to the FIFO buffer, a data packet can be transmitted in the first frame after the SOF packet is detected. This isochronous transfer transmit data setup function can identify the frame that started transmission.

When the double buffering is used, transmission is only enabled for the buffer where data writing was completed first, even after the data write to both buffers is complete. Accordingly, even if multiple IN tokens are received, only the one packet of FIFO buffer data is transmitted.

When the FIFO buffer is ready to transmit data when an IN token is received, the data is transferred and a normal response is returned. However, if the FIFO buffer cannot transmit data, a zero-length packet is transmitted and an underrun error occurs.

Figure 29.20 shows an example transmission using the isochronous transfer transmission data setup function when IITV = 0 (every frame) is set.

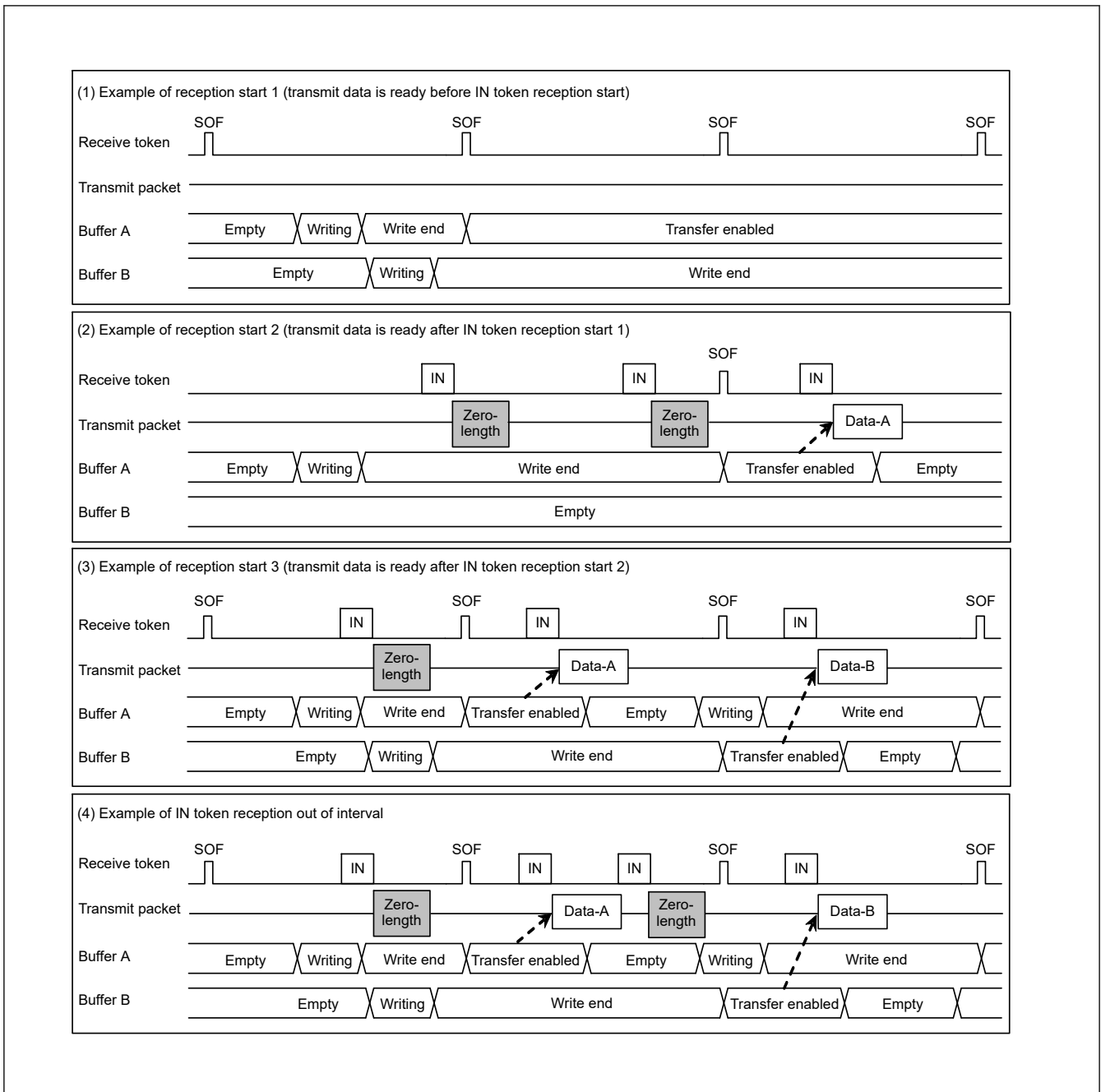


Figure 29.20 Example data setup operation

(5) Transmit buffer flush for isochronous transfers in device controller mode

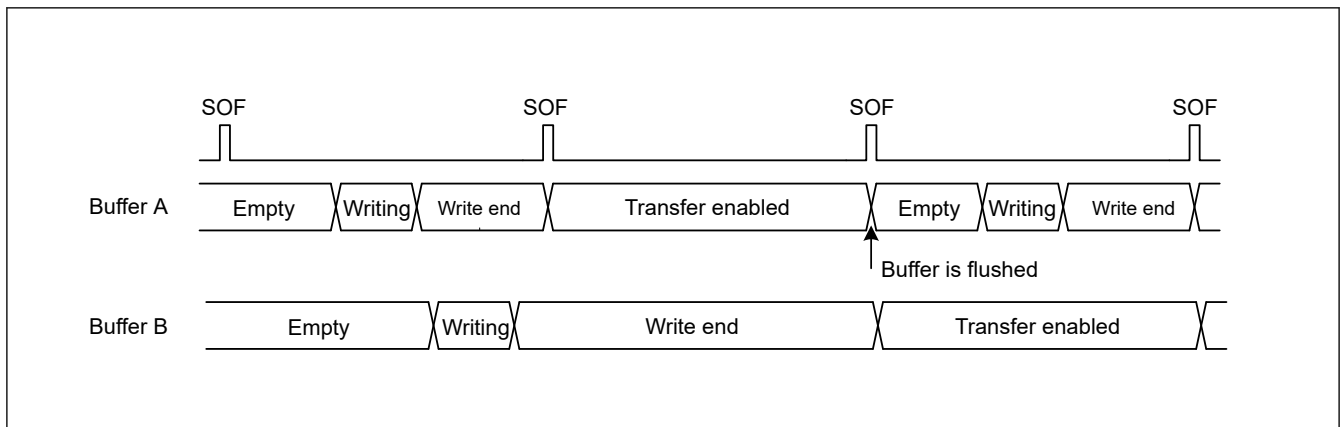
In device controller mode during isochronous data transmission, if the USBFS receives an SOF packet for the next frame without receiving an IN token in the interval frame, it operates as if the IN token is corrupt and clears the buffer that is enabled for transmission, putting that buffer in the writing enabled state.

When double buffering is used and writing to both buffers is complete, the cleared FIFO buffer is assumed to be the one where the data was transmitted in the interval frame, and transmission is enabled for the FIFO buffer that was not cleared on SOF packet reception.

The timing of the buffer flush function depends on the PIPEPERI.IITV[2:0] setting as follows:

- When IITV = 0:
  - The buffer flush operation starts from the first frame after the pipe is enabled.
- When IITV ≠ 0:
  - The buffer flush operation starts after the first normal transaction.

Figure 29.21 shows an example buffer flush. When an unanticipated token is received before the interval frame, the USBFS sends the write data or a zero-length packet as an underrun error, depending on the data setup status.



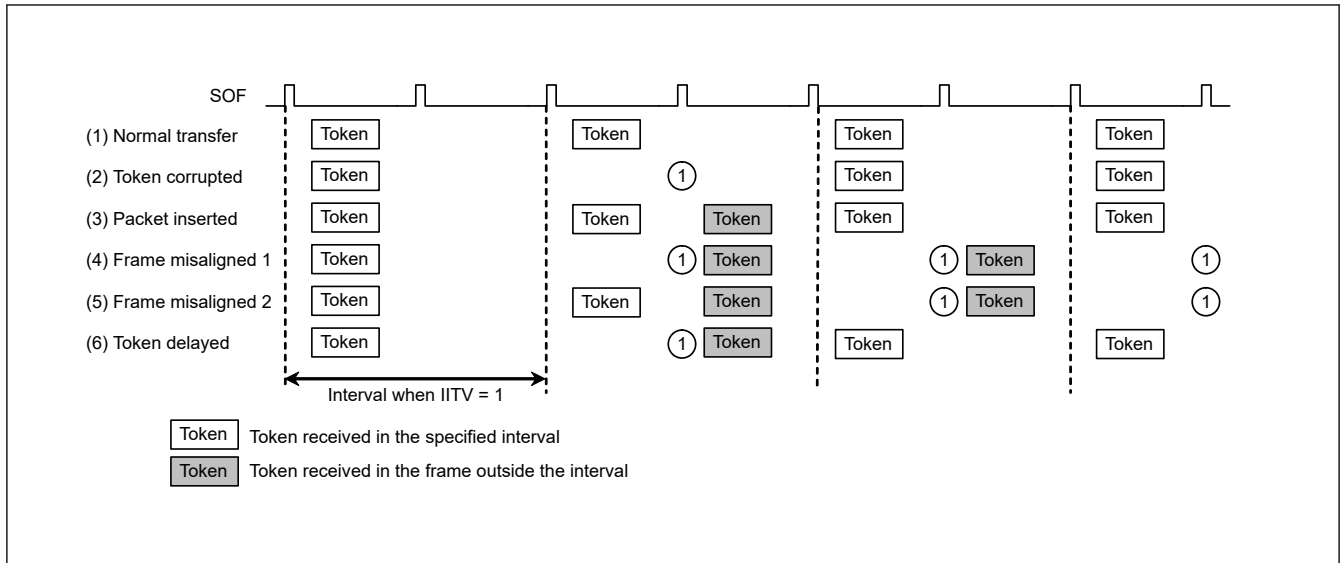
**Figure 29.21 Example buffer flush operation**

Figure 29.22 shows an example interval error occurrence. There are five types of interval errors, as shown in the figure. An interval error occurs at timing (A), and the buffer flush function is activated.

If an interval error occurs during an IN transfer, the buffer flush function is activated. If it occurs during an OUT transfer, an NRDY interrupt is generated. Use the FRMNUM.OVRN bit to distinguish between this and NRDY interrupts triggered by received packet errors and overrun errors.

For tokens that are shaded in the figure, responses are returned based on the FIFO buffer status.

- IN direction:
  - If the buffer is ready to transfer data, the data is transferred and a normal response is returned
  - If the buffer is not ready to transfer data, a zero-length packet is transmitted and an underrun error occurs
- OUT direction:
  - If the buffer is ready to receive data, the data is received and a normal response is returned
  - If the buffer is not ready to receive data, the received data is discarded and an overrun error occurs



**Figure 29.22 Example interval error occurrence when IITV = 1**

### 29.3.13 SOF Interpolation Function

In device controller mode, if packet reception is disabled at intervals of 1 ms because the SOF packet is corrupted or missing, the USBFS interpolates the SOF. SOF interpolation begins when the USBE and SCKE bits in SYSCFG are set to 1 and an SOF packet is received.

The interpolation function is initialized under the following conditions:

- MCU reset
- USB bus reset
- Suspend state detection

The SOF interpolation operates as follows:

- The interpolation function is not activated until an SOF packet is received
- When the first SOF packet is received, interpolation is performed by counting 1 ms on the 48-MHz internal clock
- When the second and subsequent SOF packets are received, interpolation is performed at the previous reception interval
- Interpolation is not performed in the Suspend state or on reception of a USB bus reset

The USBFS supports the following functions controlled by SOF packet reception. These functions operate normally with SOF interpolation if the SOF packet is missing:

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing during full-speed operation, the FRMNUM.FRNM[10:0] bits are not updated.

### 29.3.14 Pipe Schedule

#### 29.3.14.1 Conditions for generating transactions

In host controller mode and when the DVSTCTR0.UACT bit is set to 1, the USBFS generates transactions under the conditions shown in [Table 29.29](#).

**Table 29.29** Conditions for generating transactions

Transaction	Conditions for generation				
	DIR	PID	IITV0	Buffer state	SUREQ
Setup	—*1	—*1	—*1	—*1	1 setting
Control transfer data stage, status stage, bulk transfer	IN	BUF	Invalid	Receive area exists	—*1
	OUT	BUF	Invalid	Transmit data exists	—*1
Interrupt transfer	IN	BUF	Valid	Receive area exists	—*1
	OUT	BUF	Valid	Transmit data exists	—*1
Isochronous transfer	IN	BUF	Valid	*2	—*1
	OUT	BUF	Valid	*3	—*1

Note 1. An em dash (—) in the table indicates that the condition is unrelated to the generating of tokens. "Valid" indicates that, for interrupt transfers and isochronous transfers, a transaction is generated only in transfer frames that are based on the interval counter. "Invalid" indicates that a transaction is generated regardless of the interval counter.

Note 2. This indicates that a transaction is generated regardless of whether there is a receive area. If there is no receive area, however, the received data is discarded.

Note 3. This indicates that a transaction is generated regardless of whether there is any data to be transmitted. If there is no data to be transmitted, however, a zero-length packet is transmitted.

### 29.3.14.2 Transfer schedule

This section describes the transfer scheduling within a frame of the USBFS. After the USBFS sends an SOF, the transfer is carried out in the following sequence:

#### 1. Execution of periodic transfers:

A pipe is searched for in the order of pipe 1 → pipe 2 → pipe 6 → pipe 7 → pipe 8 → pipe 9, and then if there is a pipe for which an isochronous or interrupt transfer transaction can be generated, the transaction is generated.

#### 2. Setup transactions for control transfers:

The DCP is checked, and if a setup transaction is possible, it is sent.

#### 3. Execution of bulk transfers, control transfer data stages, and control transfer status stages:

A pipe is searched for in the order of DCP → pipe 1 → pipe 2 → pipe 3 → pipe 4 → pipe 5, and then if there is a pipe for which a transaction for a bulk transfer, a control transfer data stage, or a control transfer status stage can be generated, the transaction is generated.

When a transaction is generated, processing moves to the next pipe transaction regardless of whether the response from the peripheral device is ACK or NAK. If there is time for transfer within the frame, step 3 is repeated.

### 29.3.14.3 Enabling USB communication

Setting the DVSTCTR0.UACT bit to 1 initiates an SOF transmission, and transaction generation is enabled. Setting the UACT bit to 0 stops SOF transmission and the Suspend state is invoked. If the UACT setting is changed from 1 to 0, processing stops after the next SOF is sent.

## 29.4 Usage Notes

### 29.4.1 Settings for the Module-Stop State

USBFS operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The USBFS is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

### 29.4.2 Clearing the Interrupt Status Register on Canceling Software Standby Mode

Because the input buffer is always enabled in Software Standby mode, an unexpected interrupt might occur under the following conditions:

- When the interrupt is enabled in Normal mode
- When the interrupt is disabled in Software Standby mode
- When the input level of the pin that cancels software standby mode is changed in Software Standby mode

These conditions might cause the associated interrupt flag in the Interrupt Status Register to set unexpectedly. After the MCU cancels the Software Standby mode, the unexpected interrupt might be sent to the interrupt controller. To avoid this, always clear the INTSTS0 and INTSTS1 registers in the canceling sequence.

### 29.4.3 Clearing the Interrupt Status Register after Setting Up the Port Function

The input buffer is disabled before the PmnPFS.PSEL and PmnPFS.PMR port is set up, so the internal signal is fixed high or low. The input buffer is enabled after the port is set so that the external pin state is propagated to the MCU. An unexpected interrupt might occur at this time, causing the VBINT and OVRCCR bits in INTSTS0 and INTSTS1, or other interrupt status flags to set to 1. To avoid a malfunction, always clear the INTSTS0 and INTSTS1 registers after setting up the port.

### 29.4.4 Setting Up the Port Function before setting USB function

USB\_DP, USB\_DM pins are compatible with P814, P815 I/O ports. So before setting USBFS registers to enable USBFS function, set register bit P814PFS.PMR and P815PFS.PMR to 1.



## 30. Serial Communications Interface (SCI)

This is the SCI\_B version of the SCI peripheral module.

SCI\_B is referred to as SCI in this chapter.

### 30.1 Overview

The Serial Communications Interface (SCI) × 6 channels have asynchronous and synchronous serial interfaces:

- Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA))
- 8-bit clock synchronous interface
- Simple IIC (master-only)
- Simple SPI
- Simple LIN
- Smart card interface
- Manchester interface

The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. SCIn (n = 0 to 4, 9) has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using an on-chip baud rate generator.

In this section, PCLK refers to PCLKA.

Table 30.1 lists the SCI specifications, Table 30.2 shows the functions of each SCI channel, Figure 30.1 shows a block diagram of SCI, Figure 30.2 shows a clock source selector block diagram, and Table 30.3 lists the I/O pins.

**Table 30.1 SCI specifications (1 of 4)**

Parameter	Specifications
Number of modules	6 (SCIn (n = 0 to 4, 9))
Serial communication modes	<ul style="list-style-type: none"> <li>• Asynchronous</li> <li>• Clock synchronous</li> <li>• Simple IIC</li> <li>• Simple SPI</li> <li>• Simple LIN (SCIn (n = 0, 1))</li> <li>• Smart card interface</li> <li>• Manchester interface (SCIn (n = 0))</li> </ul>
Operation clock (TCLK)	Synchronized clock (PCLK) or independent clock (SCICLK) can be selected
Transfer speed	Bit rate specifiable with the on-chip baud rate generator
Full-duplex communications	<ul style="list-style-type: none"> <li>• Transmitter: Continuous transmission possible using double-buffering</li> <li>• Receiver: Continuous reception possible using double-buffering</li> </ul>
Half-duplex communications	Half-duplex communication is possible by using only TXDn pins
Data transfer	Selectable as LSB-first or MSB-first transfer
Inverter for communication pins (RXDn, TXDn)	Selectable inverter for each pins (RXDn, TXDn)
Interrupt sources	Transmit end, transmit data empty, receive data full, receive error, receive data ready, address match. Break Field detection/output, Bus collision detection, Active edge detection. (SCIn (n = 0, 1)). Completion of generation of a start condition, restart condition, or stop condition. (for simple IIC mode)
Loop Back function	Self-diagnosis of communication function by SCI internal transmission / reception is possible
Synchronizer Bypass function	Ability to bypass synchronization circuit between bus clock (PCLK) and operation clock (TCLK)
Module-stop function	Module-stop state can be set for each channels to reduce power consumption
Reception in Software Standby mode	Only SCIO is available
Clock synchronous mode	Data length
	8 bits

Table 30.1 SCI specifications (2 of 4)

Parameter		Specifications
	Adjustment of receive sampling timing	Adjustable receive sampling timing after the default timing in master mode only when using internal clock
	Receive error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pins
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Adjustment of receive sampling timing	Adjustable receive sampling timing before/after the default timing
	Adjustment of transmit timing	Adjustable edge timing of transmit waveform controlled by the setting value of registers.
	Parity	Even parity, odd parity, or no parity
	Receive error detection	<ul style="list-style-type: none"> <li>• Parity error</li> <li>• Overrun error</li> <li>• Framing error</li> </ul>
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pin and CTSn pin
	Transmission and reception	Selectable to 1-stage register or 16-stage FIFO
	Address match	Interrupt request/event output can be issued upon detecting a match between received data and the value in the compare match register
	Start-bit detection	Selectable to low level or falling edge detection
	Break detection	Breaks from framing errors detectable by read from CSR register
	Clock source	Selectable to internal or external clock. Transfer rate clock input from the GPT can be used. (SCI1)
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Multi-processor communications function	Serial communication enabled among multiple processors
	RS-485 driver control function	Output DEN signal to enable external transceiver transmit mode
Noise cancellation	Digital noise filters included on signal paths from the RXDn pin inputs	
Smart card interface mode	Error processing	Error signal can be automatically transmitted upon detecting a parity error during reception
		Data can be automatically retransmitted upon receiving an error signal during transmission
Data type	Both direct and inverse convention supported	
Manchester mode	Communication format	Manchester code with the preface and the Start Bit added
	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity function	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, framing, Manchester errors
	Hardware flow control	Transmission and reception controllable with CTSn_RTSn pin and CTSn pin
	Clock source	Only internal clock can be used.
Double-speed mode	Baud rate generator double-speed mode is selectable	

Table 30.1 SCI specifications (3 of 4)

Parameter		Specifications
	Multi-processor communication function	Serial communication among multiple processors
	Manchester encoding / decoding function	Function to perform Manchester encoding / decoding of transmission / reception data and communicate using Manchester code
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters
	Preface setting / detection function	The function outputs the configured the preface pattern and detects it.
	Start Bit setting / detection function	The function outputs the configured the Start Bit pattern and detects it.
	Reception retiming function	Timing correction is performed for each bit of the received signal
Simple IIC mode	Transfer format	I <sup>2</sup> C bus format (MSB-first only)
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 400 kbps
	Noise cancellation	The signal paths from input on the SCLn and SDAn pins incorporate digital noise filters and provide an adjustable interval for noise cancellation
Simple SPI mode	Data length	8 bits
	Error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Double-speed mode	Baud rate generator double-speed mode is selectable
	Transmission / Reception	Selectable either 1 stage register or 16-stage FIFO
	Adjustment of receive sampling timing	Adjustable receive sampling timing after the default timing in master mode only when using internal clock
	SSn input pin function	High impedance state can be invoked on the output pins by driving the SSn pin high.
	Clock settings	Configurable among four clock phase and clock polarity settings
Simple LIN mode	Start Frame Transmission	<ul style="list-style-type: none"> <li>Break Field output possible, Break Field output complete interrupt output possible</li> <li>Bus collision detection possible, bus collision detection interrupt output possible</li> </ul>
	Start Frame Reception	<ul style="list-style-type: none"> <li>Break Field detectable, Break Field detected interrupt output possible</li> <li>Control Field 0/1 data comparison function</li> <li>Control Field 1 can set two types of comparison data of primary and secondary</li> <li>Priority interrupt bit can be set in Control Field 1</li> <li>Handling of Start Frames that do not include a Break Field</li> <li>Handling of Start Frames that do not include a Control Field 0</li> <li>Bit rate measurement function</li> </ul>
	Input/Output control function	<ul style="list-style-type: none"> <li>Selectable polarity for TXDn and RXDn signals</li> <li>Selection of a digital filter for the RXDn signal</li> <li>Half-duplex operation employing RXDn and TXDn signals multiplexed on the same pin</li> <li>Selectable timing for the sampling of data received through RXDn</li> </ul>
Bit rate modulation function		Error reduction through correction of outputs from the on-chip baud rate generator
Event link function		Error event output for receive error or error signal detection (SCIn_ERI) (n = 0 to 4, 9)
		Receive data full event output (SCIn_RXI) (n = 0 to 4, 9)
		Transmit data empty event output (SCIn_TXI) (n = 0 to 4, 9)
		Address match event output (SCIn_AM) (n = 0 to 4, 9)
		Active edge detection event output (SCIn_AED) (n = 0, 1)
		Transmit end event output (SCIn_TEI) (n = 0 to 4, 9)

**Table 30.1 SCI specifications (4 of 4)**

Parameter	Specifications
TrustZone Filter	Security and Privilege attribution can be set for each channels

**Table 30.2 Functions of each SCI channel**

Item	SCI0	SCI1	SCI2	SCI3	SCI4	SCI9
Asynchronous mode	Available					
Clock synchronous mode	Available					
Smart card interface mode	Available					
Simple IIC mode	Available					
Simple SPI mode	Available					
FIFO mode	Available (16-stage)					
Address match	Available					
Manchester mode	Available	Not Available	Not Available	Not Available	Not Available	Not Available.
Simple LIN mode	Available	Available	Not Available	Not Available	Not Available	Not Available
GPT clock input	Not Available	GTIOC3A GTIOC4A	Not Available	Not Available	Not Available	Not Available

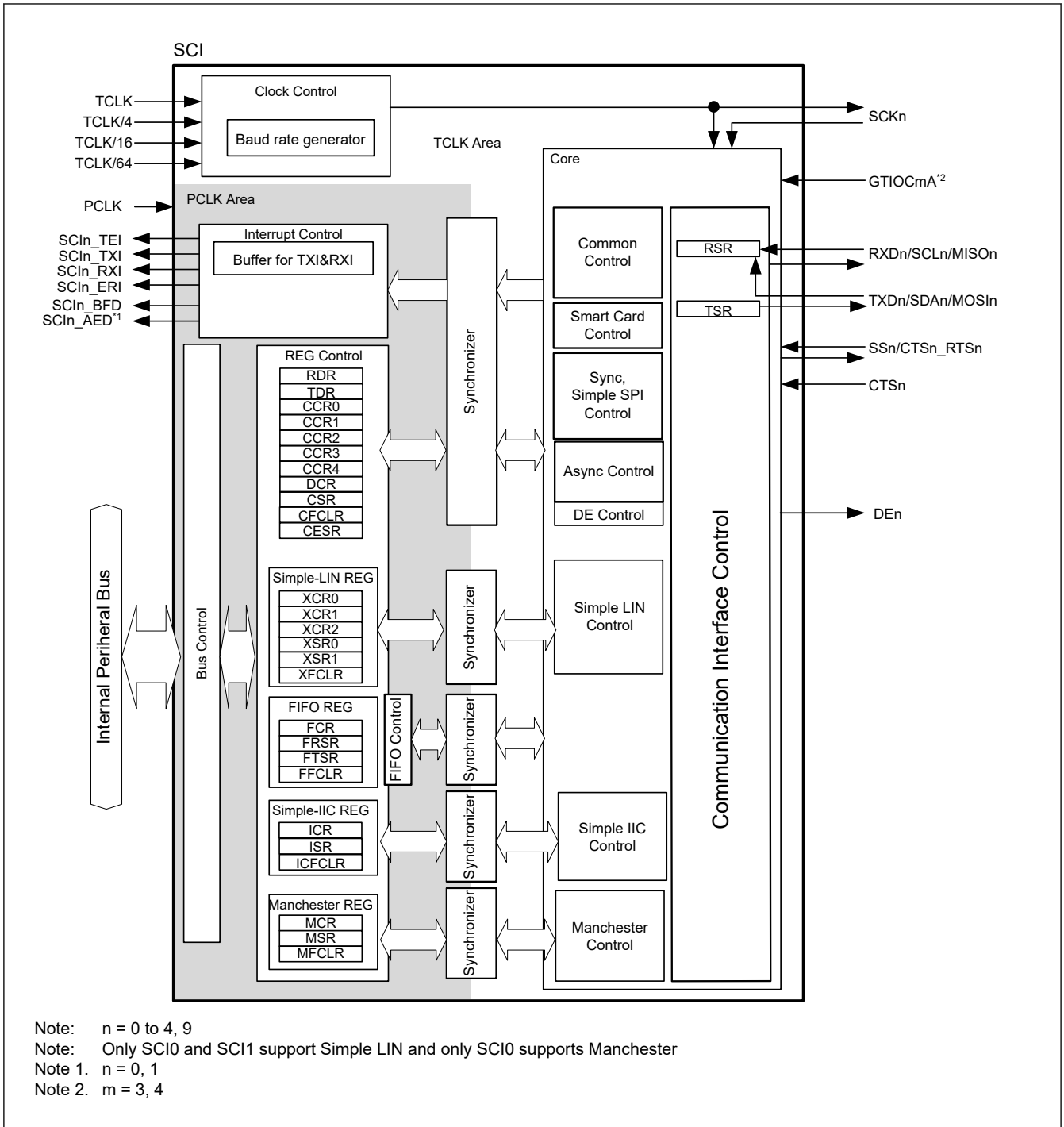


Figure 30.1 SCI block diagram

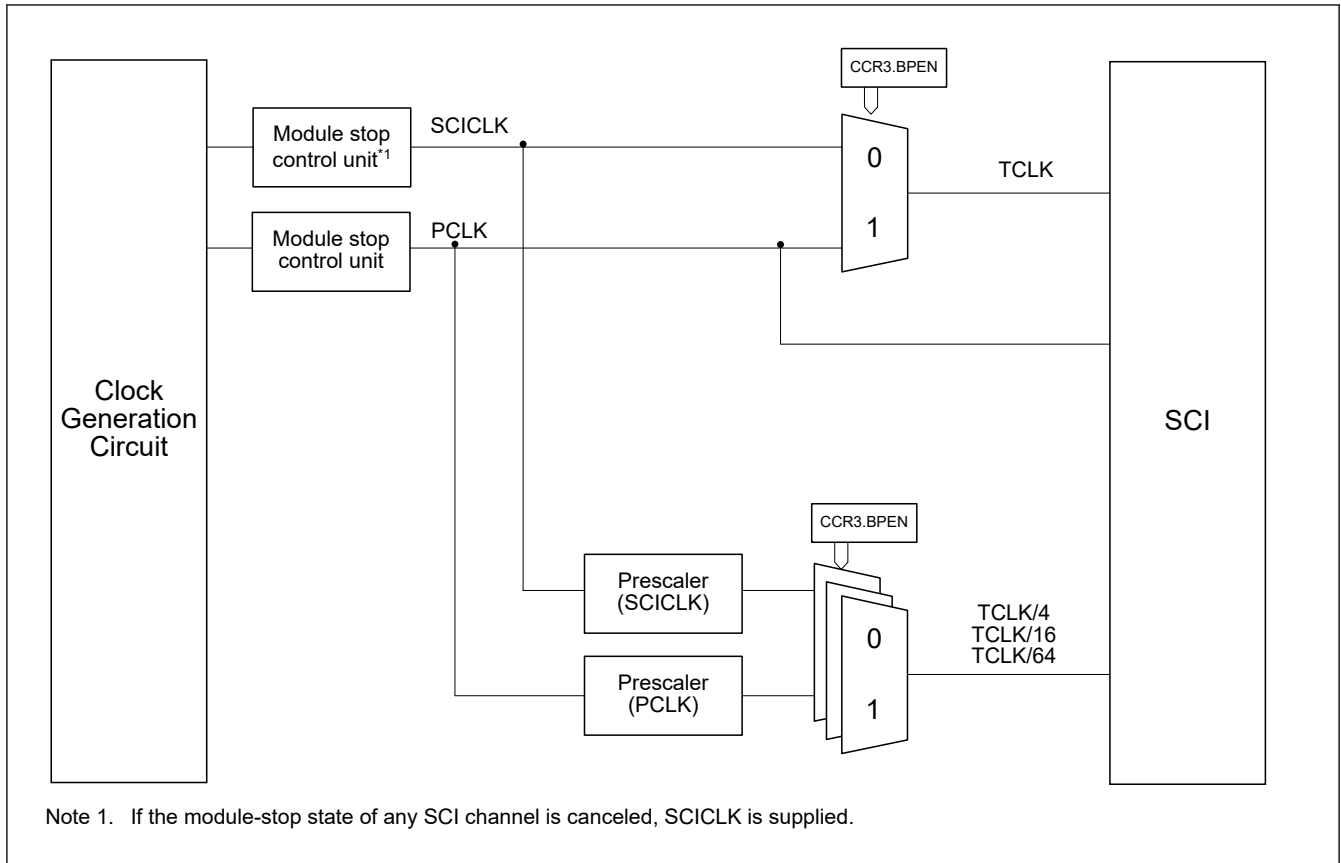


Figure 30.2 Clock source selector block diagram

Table 30.3 SCI I/O pins

Function	Pin name	Input/Output	Description
SCIn (n = 0 to 4, 9)	RXDn/SCLn/MISO <sub>n</sub>	Input/Output	SCIn receive data input SCIn I <sup>2</sup> C clock input/output SCIn slave transmit data input/output
	TXDn/SDAn/MOS <sub>n</sub>	Input/Output	SCIn transmit data output SCIn I <sup>2</sup> C data input/output SCIn master transmit data input/output
	SSn/CTS <sub>n</sub> _RTSn	Input/Output	SCIn chip select input, active-low SCIn transfer start control input/output, active-low
	CTS <sub>n</sub>	Input	SCIn transfer start control input, active-low
	DE <sub>n</sub>	Output	Driver Enable signal output
	SCK <sub>n</sub>	Input/Output	SCIn clock input/output

## 30.2 Register Descriptions

### 30.2.1 RSR : Receive Shift Register

RSR is a shift register that receives serial data input from the RXD<sub>n</sub> pin and converts it into parallel data. When one frame of data is received, the data is automatically transferred to the RDR register. The RSR register cannot be directly accessed by the CPU.

### 30.2.2 RDR/RDR\_BY : Receive Data Register

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	FER	PER	—	—	ORER	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	FFER	FPER	DR	MPB	RDAT[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	RDAT[8:0]	Serial receive data RDAT is a 9-bit register for storing received data. Received data is stored in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. And 0 is stored in the unused bit. Use RDR_BY for byte access to RDR[7:0].	R
9	MPB	Multi-processor flag 0: Data transmission cycles 1: ID transmission cycles	R
10	DR	Receive data ready flag FRSR.DR can be read.	R
11	FPER	FIFO parity error flag Valid only in Asynchronous mode 0: There is no parity error in the data read from the receive-FIFO 1: There is parity error in the data read from the receive-FIFO	R
12	FFER	FIFO framing error flag Valid only in Asynchronous mode 0: There is no framing error in the data read from the receive-FIFO 1: There is framing error in the data read from the receive-FIFO	R
23:13	—	These bits are read as 0.	R
24	ORER	Overrun Error flag CSR.ORER can be read.	R
26:25	—	These bits are read as 0.	R
27	PER	Parity error flag CSR.PER can be read.	R
28	FER	Framing error flag CSR.FER can be read.	R
31:29	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

In FIFO mode (CCR3.FM = 1), the RDR/RDR\_BY is 16-stage FIFO buffer configuration.

When using FIFO mode, use RDR for 32-bit access.

#### RDAT[8:0] bits (Serial receive data)

After one frame of data is received, the received data is transferred from the RSR register to this register, thus allowing the RSR register to receive the next data.

The RSR and RDR registers have a double-buffered construction to enable continuous reception.

For Non-FIFO mode, read RDR only once when a receive data full interrupt (SCIn\_RXI) request is issued. Without reading received data from RDR, if the next one frame is received, an overrun error occurs.

For FIFO mode, continuous reception is executed until 16 stages are stored. If data is read when there is no received data in the receive-FIFO(RDR), the value is undefined. When the receive-FIFO (RDR) are full of received data, subsequent serial receive data is lost.

The CPU cannot write to RDR.

0 is stored in the bit position which isn't received (RDR.bit8 or RDR.bit7) at the time of 7bit or 8bit communication of Asynchronous and Manchester mode.

**MPB bit (Multi-processor flag)**

In Asynchronous mode and Manchester mode, during multi-processor communication (CCR3.MP = 1), the value of the multi-processor bit corresponding to the received data (RDAT[8:0]) can be read.

**FPER bit (FIFO parity error flag)**

Indicates whether the data read from the receive-FIFO has a parity error.

0 is stored for non-FIFO mode.

**FFER bit (FIFO framing error flag)**

Indicates whether the data read from receive-FIFO has a framing error.

0 is stored for non-FIFO mode.

**30.2.3 TDR/TDRLL/TDRLH : Transmit Data Register**

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x04(TDR/TDRLL)  
 0x05(TDRLH)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TSYN C	—	—	MPBT	TDAT[8:0]								
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
8:0	TDAT[8:0]	Serial transmit data TDAT is a 9-bit register for setting transmit data. Transmit data is set in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. When byte access, write TDR [15:8] and then write TDR [7:0]. Use TDRLL for byte access to TDR[7:0], TDRLH for byte access to TDR[15:8].	R/W
9	MPBT	Multi-processor transfer bit flag Value of the multi-processor bit in the transmission frame. This bit is use in Asynchronous and Manchester mode. When writing to this bit when not used, write the initial value. 0: Data transmission cycles 1: ID transmission cycles	R/W
11:10	—	These bits are read as 1. The write value should be 1.	R/W
12	TSYNC	Transmit SYNC data It is valid when MCR.SBSEL = 1 and MCR.SYNSEL = 1 in Manchester mode. When this bit is not used, write the initial value. 0: The Start Bit is transmitted as DATA SYNC. 1: The Start Bit is transmitted as COMMAND SYNC.	R/W
31:13	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-3, P-TYPE-3

In FIFO mode (CCR3.FM = 1), the TDR/TDRLL/TDRLH is 16-stage FIFO buffer configuration.



When using FIFO mode, use TDR for 32-bit access.

### TDAT[8:0] bits (Serial transmit data)

The is a 9-bit register for storing transmit data.

When empty space is detected in the TSR register, the transmit data stored in the TDR/TDRLL/TDR LH registers is transferred to TSR, and transmitting is started.

The TSR and TDR/TDRLL/TDR LH registers have a double-buffered construction to realize continuous transmission. When the next data to be transmitted is stored in TDR/TDRLL/TDR LH after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

When the SCI detects that the transmit shift register (TSR) is empty, it transmits data written in the transmit-FIFO (TDR/TDRLL/TDR LH) into TSR and starts serial transmission. Continuous serial transmission is executed until there is no transmit data left in the transmit-FIFO (TDR/TDRLL/TDR LH).

For non-FIFO mode, when a transmit data empty interrupt (SCIn\_TXI) request is issued and CCR0.TE is 1, write transmit data to the TDR only once.

For FIFO mode, when transmit-FIFO is full of transmit data 16 frames, no more data can be written. If writing of new data is attempted, the data is ignored.

The TDR/TDRLL/TDR LH register can always be read / written from the CPU. And when byte access, write TDR/TDRLL/TDR LH[15:8] and then write TDR/TDRLL/TDR LH[7:0].

### MPBT bit (Multi-processor transfer bit flag)

Selects the multi processor bit of transmit frame.

### TSYNC bit (Transmit SYNC data)

When Manchester mode and MCR.SBSEL = 1 and MCR.SYNSEL = 1, the type of SYNC selected according to the TSYNC bit becomes the Start Bit of the transmission frame.

## 30.2.4 TSR : Transmit Shift Register

TSR is a shift register that transmits serial data. To perform serial data transmission, the SCI first automatically transfers transmit data from TDR to TSR, then sends the data to the TXDn pin. The CPU cannot directly access the TSR.

## 30.2.5 CCR0 : Common Control Register 0

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	SSE	—	—	TEIE	TIE	—	—	—	—	RIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IDSEL	DCME	MPIE	—	—	—	TE	—	—	—	RE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RE	Receive Enable 0: Serial reception is disabled 1: Serial reception is enabled	R/W <sup>*1</sup> *3
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	TE	Transmit Enable 0: Serial transmission is disabled 1: Serial transmission is enabled	R/W <sup>*1</sup>

Bit	Symbol	Function	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	MPIE	Multi-Processor Interrupt Enable Valid in Asynchronous mode and Manchester mode when CCR3.MP is 1. This bit should set 0 in Smart card interface mode.  0: Non-multi-processor reception 1: Multi-processor reception When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and non-multi-processor reception is resumed. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame (consider the synchronization delay time).	R/W <sup>2</sup>
9	DCME	Data Compare Match Enable Valid only in Asynchronous mode  0: Address match function is disabled 1: Address match function is enabled	R/W <sup>2</sup>
10	IDSEL	ID Frame Select Valid only in Asynchronous mode with multi-processor  0: Compare data irrespective of the value of the MPB bit 1: Compare data only when the MPB bit is 1 (ID frame)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	RIE	Receive Interrupt Enable  0: SCIn_RXI and SCIn_ERI interrupt requests are disabled 1: SCIn_RXI and SCIn_ERI interrupt requests are enabled	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TIE	Transmit Interrupt Enable  0: SCIn_TXI interrupt request is disabled 1: SCIn_TXI interrupt request is enabled	R/W
21	TEIE	Transmit End Interrupt Enable This bit should set 0 in Smart card interface mode.  0: SCIn_TEI interrupt request is disabled 1: SCIn_TEI interrupt request is enabled	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	SSE	SSn Pin Function Enable Valid in Simple SPI mode. In slave mode (CCR3.CKE[1:0] = 1x), set 1 to this bit.  0: SSn pin function is disabled 1: SSn pin function is enabled	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. In Clock synchronous mode (CCR3.MOD[2:0] = 010b), Simple SPI mode (CCR3.MOD[2:0] = 011b), and Simple IIC mode (CCR3.MOD[2:0] = 100b), 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE. In other mode, writing is enabled under any condition.

Note 2. This bit is a bit that is cleared by hardware. Note that writing to a bit other than this bit with a bit manipulation instruction may cause this bit to be unintentionally set to 1 by a read-modify-write operation.

Note 3. In Clock synchronous mode or Simple SPI mode, and internal clock (master mode), receive-only setting is prohibited in the internal clock (master mode) (TE = 0 and RE = 1 setting prohibited).

### RE bit (Receive Enable)

Enables or disables serial receive operation.

When the RE bit is set to 1, serial reception becomes possible after the synchronization delay time has elapsed in Asynchronous mode or the synchronous clock input in Clock synchronous mode or the neg-edge of RXDn in Manchester mode or start bit in Smart card interface mode.

Note that CCR3 should be set prior to setting the RE bit to 1 in order to designate the reception format.

Except Smart card interface mode, even if reception is halted by setting the RE bit to 0, the CSR.RDRF, FER, PER, ORER, MSR.MER, SBER, SYER, PFER, FRSR. DR flags are not affected, and the previous values is retained. In Smart card interface mode, even if reception is halted by setting the RE bit to 0, the CSR.FER, PER, ORER flags are not affected and

the previous value is retained. Also, to stop the reception operation, synchronization delay time is required from when the RE bit is set to 0 until the reception operation is stopped.

### TE bit (Transmit Enable)

Enables or disables serial transmission.

When the TE bit is set to 1, serial transmission becomes possible after the synchronization delay time has elapsed. After the synchronization delay time, transmission is started by writing transmit data to TDR. Note that CCR3 should be set prior to setting the TE bit to 1 in order to designate the transmission format. In addition, the synchronization delay time is required until the transmission control circuit is stopped after the TE bit is set to 0.

### MPIE bit (Multi-Processor Interrupt Enable)

When the MPIE bit is set to 1 and data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags (CSR.RDRF, ORER, FER, FRSR, DR, MSR.MER, SYER, PFER, SBER) are disabled.

When data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, see [section 30.4. Multi-Processor Communication Function](#). If you want to continue receiving operation using the multi-processor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame.

When the receive data includes the MPB bit set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER, FER, MER, SYER, PFER, and SBER to 1 is disabled.

When the receive data includes the MPB bit set to 1, the MPIE bit is automatically cleared to 0, the SCIn\_RXI and SCIn\_ERI interrupt requests are enabled (if CCR0.RIE is set to 1), and setting the flags ORER, FER, MER, SYER, PFER, and SBER to 1 is enabled.

MPIE should be set to 0 if multi-processor communications function is not to be used.

### DCME bit (Data Compare Match Enable)

The DCME bit selects whether the address match function (data compare match function) is enabled or not.

When DCME is 1, if SCI detects the match to the comparison data (CCR4.CMPD) with receive data, DCME is cleared automatically, and after that, SCI operation mode is in receive mode without data compare match function.

See [section 30.3.6. Address Match \(Receive Data Match Detection\) Function](#).

The write value should be 0 other than Asynchronous mode.

### IDSEL bit (ID Frame Select)

The IDSEL bit selects whether to compare data irrespective of the value of MPB bit or to compare data only when the MPB bit = 1 (ID frame) when the address match function is valid. Set at the same time as DCME.

### RIE bit (Receive Interrupt Enable)

Enables or disables SCIn\_RXI and SCIn\_ERI interrupt requests.

SCIn\_RXI and SCIn\_ERI interrupt request is disabled by setting the RIE bit to 0.

An SCIn\_ERI interrupt request can be canceled by reading 1 from the CSR.ORER, FER, or PER and then setting the flag to 0 or setting the RIE bit to 0.

In Manchester mode, the MER, SYER, PFER and SBER flags are also the cause of SCIn\_ERI interrupt request, so the same processing is necessary. For details of these flags, see [section 30.2.12. MCR : Manchester Control Register](#) and [section 30.2.21. MSR : Manchester Status Register](#).

### TIE bit (Transmit Interrupt Enable)

Enables or disables SCIn\_TXI interrupt request.

An SCIn\_TXI interrupt request is disabled by setting the TIE bit to 0. At the beginning of transmission, set 1 to CCR0.TE and CCR0.TIE simultaneously. The SCIn\_TXI interrupt request is then generated.

### TEIE bit (Transmit End Interrupt Enable)

Enables or disables a SCIn\_TEI interrupt request. A SCIn\_TEI interrupt request is disabled by setting the TEIE bit to 0.

In simple IIC mode, the SCIn\_TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STIn). In this case, the TEIE bit can be used to enable or disable the STIn.

### SSE bit (SSn Pin Function Enable)

Set the SSE bit to 1 if the SSn pin is to be used in control of transmission and reception (in simple SPI mode).

Set the SSE bit to 0 in any other mode. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

In the slave mode (CCR3.CKE[1:0] = 10 or 11), SSE should be set 1.

In the master mode (CCR3.CKE[1:0] = 00 or 01) and single-master, the SSn pin on the master side is not required to control reception and transmission, so SSE should be set 0.

## 30.2.6 CCR1 : Common Control Register 1

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	NFM	NFEN	—	NFCS[2:0]			—	—	—	SHAR PS	—	—	—	SPLP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RINV	TINV	—	—	PM	PE	—	—	SPB2I O	SPB2 DT	—	—	CTSP EN	CTSE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	CTSE	CTS Enable 0: CTS function is disabled (RTS output function is enabled) 1: CTS function is enabled	R/W
1	CTSPEN	CTS External Pin Enable 0: Alternate setting to use CTS and RTS functions as either 1 pin 1: Dedicated setting for separately using CTS and RTS functions with 2 pins	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	SPB2DT	Serial Port Break Data Select The output level of TXDn pin is selected when CCR0.TE = 0 and SPB2IO = 1.*1 0: When TINV is 0, low level is output in TXDn pin. When TINV is 1, high level is output in TXDn pin. 1: When TINV is 0, high level is output in TXDn pin. When TINV is 1, low level is output in TXDn pin.	R/W
5	SPB2IO	Serial Port Break I/O This bit selects whether the value of SPB2DT is output to TXDn pin when CCR0.TE = 0.*1 0: The value of SPB2DT bit is not output in TXDn pin 1: The value of SPB2DT bit is output in TXDn pin	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	PE	Parity Enable Valid only in Asynchronous mode and Manchester mode. In Smart Card Interface mode, set 1 to this bit. 0: When transmitting: Do not add parity bit When receiving: Do not check parity bit 1: When transmitting: Add parity bit When receiving: Check parity bit	R/W
9	PM	Parity Mode Valid only when the PE bit is 1 0: Selects even parity 1: Selects odd parity	R/W

Bit	Symbol	Function	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	TINV	TXD Invert 0: Transmit data is not inverted and output to TXDn*2 1: Transmit data is inverted and output to TXDn	R/W
13	RINV	RXD Invert 0: Received data from RXDn is not inverted and input*2 1: Received data from RXDn is inverted and input	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	SPLP	Loopback Control This bit can be used for internal clock operation in Asynchronous mode, internal mode operation in Manchester mode, internal clock operation in Clock synchronous mode. 0: Normal mode 1: Loopback mode	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	SHARPS	Half-Duplex Communication Select In Simple IIC mode, Smart Card Interface mode, or Simple SPI mode, this bit should be set to 0. 0: TXDn pin, RXDn pin independent 1: TXDn/RXDn pin combination use (half-duplex communication using TXDn pin)	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
26:24	NFCS[2:0]	Noise Filter Clock Select Valid in Asynchronous mode and Manchester mode, Simple LIN mode, and Simple IIC mode. In Simple IIC mode, 000 setting is prohibited. The on-chip baud rate generator source clock means the clock selected by CCR2.CKS [1:0]. Select for the clock source of the noise filter. 0 0 0: The base clock signal divided by 1 0 0 1: The on-chip baud rate generator source clock divided by 1 0 1 0: The on-chip baud rate generator source clock divided by 2 0 1 1: The on-chip baud rate generator source clock divided by 4 1 0 0: The on-chip baud rate generator source clock divided by 8 others: Setting prohibited	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	NFEN	Digital Noise Filter Function Enable Valid in Asynchronous mode, Manchester mode, Simple LIN mode, and Simple IIC mode 0: In Asynchronous, Manchester, Simple LIN modes: Disable noise cancellation function for RXDn input signal In Simple IIC mode: Disable noise cancellation function for SCLn and SDAn input signals 1: In Asynchronous, Manchester, Simple LIN modes: Enable noise cancellation function for RXDn input signal In Simple IIC mode: Enable noise cancellation function for SCLn and SDAn input signals	R/W
29	NFM	Noise Filter Mode Valid in asynchronous mode and ABCSE2 = 1 0: 3-point matching mode 1: Majority vote mode	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Use this bit in Asynchronous mode and Manchester mode only. Operation in other mode is not guaranteed.

Note 2. RINV/TINV should be set to 0 in Smart card interface mode and Simple IIC mode.

### CTSE bit (CTS Enable)

Set the CTSE bit to 1 if the SSn pin is to be used for inputting of the CTSn control signal to control of transmission and reception. The RTSn signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, Simple LIN mode, and simple IIC mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

**CTSPEN bit (CTS External Pin Enable)**

When CTSE is 1, select the pins usage method when using the CTS and RTS functions. Set this bit to 1 when assigning the CTS/RTS function to 2 pins and using them at the same time. Set it to 0 except in Asynchronous and Manchester modes.

Table 30.4 shows the relationship between the CTSE bit and CTSPEN bit settings and the functions of the CTSn\_RTSn pin and CTSn pin.

**Table 30.4 CTSE bit and CTSPEN bit settings and pin functions**

CTSE bit	CTSPEN bit	CTS <sub>n</sub> _RTS <sub>n</sub> pin	CTS <sub>n</sub> pin
0	0	RTS <sub>n</sub> signal output	Not use
1	0	CTS <sub>n</sub> signal input	Not use
1	1	RTS <sub>n</sub> signal output	CTS <sub>n</sub> signal input
0	1	prohibition	prohibition

**SPB2DT bit (Serial Port Break Data Select), SPB2IO bit (Serial Port Break I/O)**

The TXD<sub>n</sub> pin status determined by combination of CCR0.TE bit, CCR1.SPB2IO bit, and CCR1.SPB2DT bit is indicated in Table 30.5.

**Table 30.5 TXD<sub>n</sub> pin status**

Value of CCR0.TE	Value of CCR1.SPB2IO	Value of CCR1.SPB2DT	TXD <sub>n</sub> pin status (when TINV is 0)
0	0	—	Hi-Z (initial value)
0	1	0	Low level output
0	1	1	High level output
1	—	—	Serial transmission data is output

Note: —: Don't care

**PE bit (Parity Enable)**

When the PE bit to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

In the multi-processor format, the parity bit is not added or checked regardless of the PE bit setting.

**PM bit (Parity Mode)**

The PM bit selects the parity mode for transmission and reception (even or odd). In Multi-processor mode, the PM bit is invalid.

For details on the usage of this bit in Smart card interface mode, see [section 30.7.2. Data Format \(Except in Block Transfer Mode\)](#).

**TINV bit (TXD Invert), RINV bit (RXD Invert)**

The data of RDR is controlled by RINV and CCR3.SINV. And the data from TXD<sub>n</sub> pin is controlled by TINV and CCR3.SINV. The control by RINV/TINV are done to communication pins (RXD<sub>n</sub> / TXD<sub>n</sub>), so they can control not only data-bits but also other bits (start bit, stop bit, parity bit). For details, see [Figure 30.3](#).

During half-duplex communication and slave operation in Simple SPI mode, use the TXD<sub>n</sub> pin for reception, so set the inversion control of the received data with the TINV bit.

Note: Description and a timing chart of the SCI operation are provided when TINV = 0 and RINV = 0 (when the value of TINV and the value of RINV are not specified).

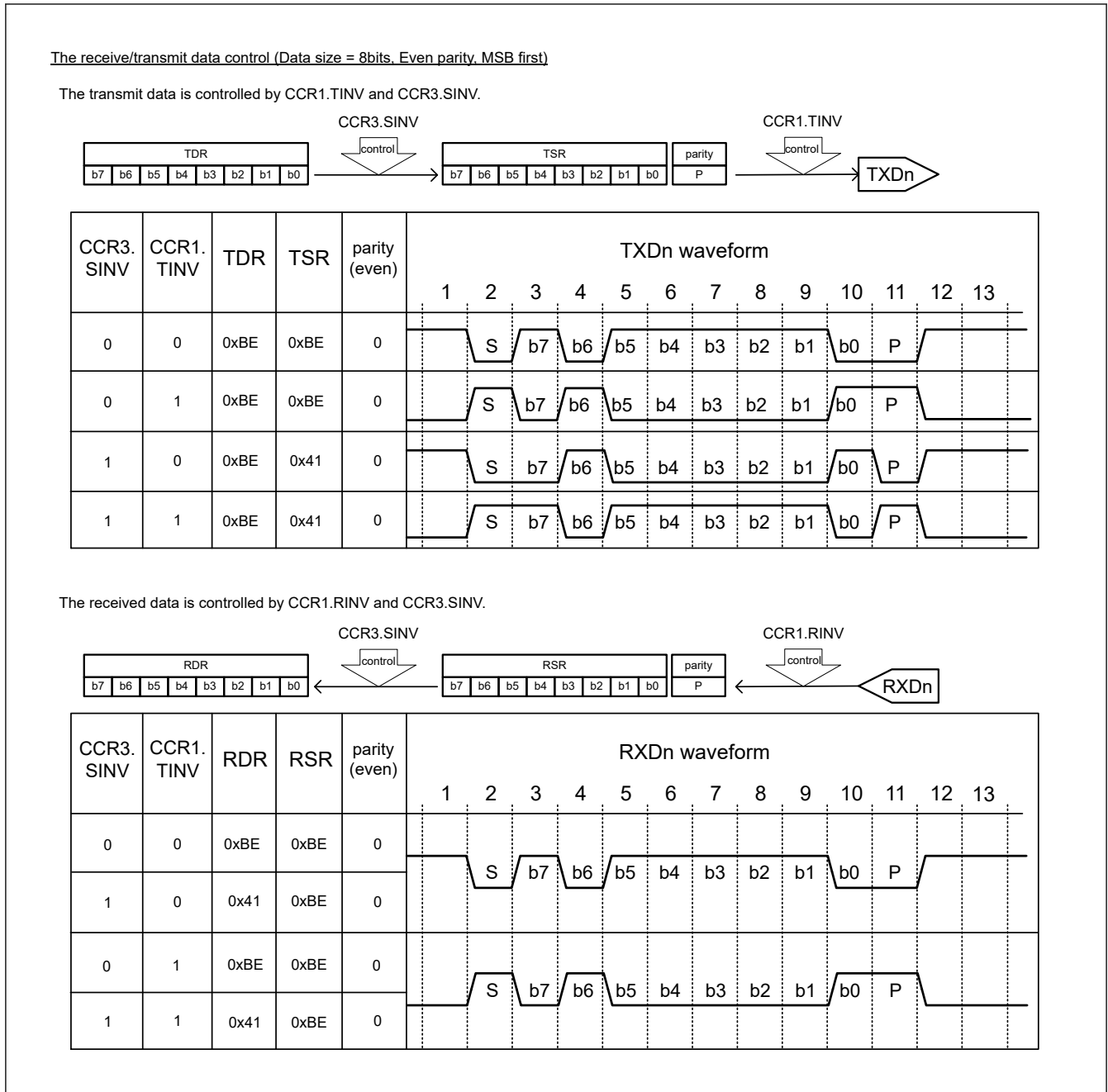


Figure 30.3 Example of receive or transmit data control

**SPLP bit (Loopback Control)**

When the SPLP bit is 1, SCI blocks the input path from RXDn and connects the output path to TXDn to the reception data register.

Transmit data can be inverted and received by combining it with TINV bit.

Set this bit to 0 for slave operation in Clock synchronous mode, when using an external clock in Asynchronous mode, and when in Simple LIN mode.

**SHARPS bit (Half-Duplex Communication Select)**

Setting the SHARPS bit to 1 enables half-duplex communication using the TXDn pin. However, the SHARPS bit cannot be used in Simple SPI mode, Simple IIC mode and Smart card interface mode.

If the SHARPS bit is set to 1 and CCR0.TE = 1, CCR0.RE = 0, the TXDn pin becomes the communication output. If this bit is set to 1 and CCR0.TE = 0, CCR0.RE = 1, the TXDn pin becomes the communication input. For details, see [section 30.17. Half-Duplex communication Function](#).

**NFCS[2:0] bits (Noise Filter Clock Select)**

The NFCS[2:0] bits select the sampling clock for the digital noise filter.

To use the noise filter in Asynchronous mode, Manchester mode, and Simple LIN mode set the NFCS[2:0] bits from 000b to 100b. In Simple IIC mode, set the bits to a value in the range from 001b to 100b. In particular, if the ABCSE or ABCSE2 bit is set to 1 in asynchronous mode, set it to 000b or 001b.

**NFEN bit (Digital Noise Filter Function Enable)**

The NFEN bit enables or disables the digital noise filter function. When the function is enabled, noise cancellation is applied to the RXDn input signal in Asynchronous mode, Manchester mode, Simple LIN mode, and noise cancellation is applied to the SDAn and SCLn input signals in Simple IIC mode. In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function. When the function is disabled, input signals are transferred as is, as internal signals.

**NFM bit (Noise Filter Mode)**

The NFM bit selects the operation mode of the digital noise filter function. For details, see [section 30.14. Noise Cancellation Function](#).

**30.2.7 CCR2 : Common Control Register 2**

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MDDR[7:0]							—	—	CKS[1:0]	—	—	—	BRME		
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BRR[7:0]							ABCS E2	ABCS E	ABCS	BGDM	—	BCP[2:0]			
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	BCP[2:0]	Base Clock Pulse Selects the number of base clock cycles in Smart card interface mode. 0 0 0: 93 clock cycles (S = 93) <sup>*1</sup> 0 0 1: 128 clock cycles (S = 128) <sup>*1</sup> 0 1 0: 186 clock cycles (S = 186) <sup>*1</sup> 0 1 1: 512 clock cycles (S = 512) <sup>*1</sup> 1 0 0: 32 clock cycles (S = 32) <sup>*1</sup> (Initial value) 1 0 1: 64 clock cycles (S = 64) <sup>*1</sup> 1 1 0: 372 clock cycles (S = 372) <sup>*1</sup> 1 1 1: 256 clock cycles (S = 256) <sup>*1</sup>	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	BGDM	Baud Rate Generator Double-Speed Mode Select Valid in Asynchronous/Manchester/Clock-synchronous/Simple SPI mode and when CCR3.CKE[1] = 0. 0: Baud rate generator outputs the clock with single frequency 1: Baud rate generator outputs the clock with doubled frequency	R/W
5	ABCS	Asynchronous Mode Base Clock Select Valid only in Asynchronous mode, Manchester mode, and Simple LIN mode 0: Selects 16 base clock cycles for 1-bit period 1: Selects 8 base clock cycles for 1-bit period	R/W



Bit	Symbol	Function	R/W
6	ABCSE	Asynchronous Mode Extended Base Clock Select Valid only in Asynchronous mode and when CCR3.CKE[1] = 0 0: Clock cycles for 1-bit period is determined by combination between CCR2.BGDM and CCR2.ABCS. 1: Baud rate is 6 base clock cycles for 1-bit period and the clock of a double frequency is output from the baud rate generator.	R/W
7	ABCSE2	Asynchronous Mode Extended Base Clock Select 2 Valid only in asynchronous mode and CCR3.CKE[1] = 0 0: Clock cycles for 1-bit period is decided with combination between CCR2.BGDM and CCR2.ABCS. 1: Baud rate is 4 base clock cycles for 1-bit period and the clock of a double frequency is output from the baud rate generator.	R/W
15:8	BRR[7:0]	Bit Rate Setting BRR is an 8-bit register that adjusts the bit rate.	R/W
16	BRME	Bit Rate Modulation Enable 0: Bit rate modulation function is disabled 1: Bit rate modulation function is enabled	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
21:20	CKS[1:0]	Clock Select 0 0: TCLK clock ( $n = 0$ )* <sup>2</sup> 0 1: TCLK/4 clock ( $n = 1$ )* <sup>2</sup> 1 0: TCLK/16 clock ( $n = 2$ )* <sup>2</sup> 1 1: TCLK/64 clock ( $n = 3$ )* <sup>2</sup>	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	MDDR[7:0]	Modulation Duty Setting MDDR corrects the bit rate adjusted by the BRR[7:0] bits.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. S is the value of S in BRR[7:0] bits explanation.

Note 2. n is the decimal notation of the value of n in BRR[7:0] bits explanation.

### BCP[2:0] bits (Base Clock Pulse)

The BCP[2:0] bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

For details, see [section 30.7.4. Receive Data Sampling Timing and Reception Margin](#).

### BGDM bit (Baud Rate Generator Double-Speed Mode Select)

The BGDM bit is valid when the on-chip baud rate generator is selected as the clock source (CCR3.CKE[1] = 0) in Asynchronous mode, Manchester mode, Clock synchronous mode, Simple SPI mode. When external clock is selected (CCR3.CKE[1] = 1), set it to 0. For the clock output from the baud rate generator, either single or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved, and the bit rate is doubled.

Set this bit to 0 in modes other than Asynchronous mode, Manchester mode, Clock synchronous mode, or Simple SPI mode.

### ABCS bit (Asynchronous Mode Base Clock Select)

Selects the clock cycles for 1-bit period.

Set the ABCS bit to 0 in modes other than Asynchronous mode, Manchester mode, or Simple LIN mode.

### ABCSE bit (Asynchronous Mode Extended Base Clock Select)

The pulse number for a base clock at 1-bit period is 6 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 6 dividing frequency of the bus clock can this bit be used to set CCR2.CKS [1:0] = 00b and BRR[7:0] = 0x00.

Set it to 0 in modes other than Asynchronous mode. Even in Asynchronous mode, set it to 0 when using external clock.

**ABCSE2 bit (Asynchronous Mode Extended Base Clock Select 2)**

The pulse number for a base clock at 1-bit period is 4 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 4 dividing frequency of the bus clock, please use this bit and set CCR2.CKS [1:0] = 00 and BRR[7:0] = 0.

Set it to 0 in modes other than asynchronous mode. Even in asynchronous mode, set it to 0 when using external clock.

**Table 30.6 Base clock cycle number per 1-bit**

ABCSE2	ABCSE	ABCS	BGDM	The base clock cycles /1bit	The frequency of the baud rate generator
0	0	0	0	16	×1
0	0	0	1	16	×2
0	0	1	0	8	×1
0	0	1	1	8	×2
0	1	— (don't care)	— (don't care)	6	×2
1	0	— (don't care)	— (don't care)	4	×2

**BRR[7:0] bit (Bit Rate Setting)**

BRR is an 8-bit register that adjusts the bit rate.

SCI has independent baud rate generator control, different bit rates can be set for each. Table 30.7 shows the relationship between the setting (N) in the BRR and the bit rate (B) for Asynchronous mode, multiprocessor transfer, Manchester mode, clock synchronous mode, smart card interface mode, simple SPI mode, and simple IIC mode.

**Table 30.7 Relationship between N Setting in BRR and Bit Rate B**

Mode	CCR2 settings				BRR[7:0] setting	Error*4
	BGDM bit	ABCS bit	ABCSE bit	ABCSE 2 bit		
Asynchronous, multi-processor, Manchester*2, Simple-LIN*3	0	0	0	0	$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error (\%) = \left\{ \frac{TCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0	0	0	$N = \frac{TCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error (\%) = \left\{ \frac{TCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	0	0		
	1	1	0	0	$N = \frac{TCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error (\%) = \left\{ \frac{TCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	Don't care	Don't care	1	0	$N = \frac{TCLK \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error (\%) = \left\{ \frac{TCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	Don't care	Don't care	0	1	$N = \frac{TCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	$Error (\%) = \left\{ \frac{TCLK \times 10^6}{B \times 8 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI	0	0 (Initial value)	0 (Initial value)	0 (Initial value)	$N = \frac{TCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	—
	1	0 (Initial value)	0 (Initial value)	0 (Initial value)	$N = \frac{TCLK \times 10^6}{4 \times 2^{2n-1} \times B} - 1$	—
Smart card interface					$N = \frac{TCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error (\%) = \left\{ \frac{TCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
Simple IIC*1					$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	—

Note: B: Bit rate (bps)  
 N: BRR setting for on-chip baud rate generator ( $0 \leq N \leq 255$ )  
 TCLK: Operating frequency (MHz)  
 n and S: Determined by the settings of the CCR2 registers as listed in [Table 30.9](#) and [Table 30.10](#). Please be careful about  $2^{(2n+1)}$  is used in the expression for Smart card interface,  $2^{(2n-1)}$  is used in other mode.

Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple IIC mode satisfy the IIC standard.

Note 2. In Manchester mode, only ABCSE = 0 and ABCSE2 = 0 can be selected.

Note 3. In Simple-LIN mode, BGDM = 0, ABCSE = 0 and ABCSE2 = 0 can be selected.

Note 4. Error means that if the N value calculated by the above formula is not an integer for the baud rate value B to be set, the BRR is set to an approximate integer value in which case it represents the error between the baud rate value and B.

**Table 30.8 Calculating widths at high and low Level for SCL**

Mode	SCLn	Formula (result in seconds)
IIC	Width at high level (minimum value)	$(N + 1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{TCLK \times 10^6}$
	Width at low level (minimum value)	$(N + 1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{TCLK \times 10^6}$

**Table 30.9 Clock source settings**

CCR2 setting	Clock source	n
CKS[1:0] bit		
00	TCLK clock	0
01	TCLK/4 clock	1
10	TCLK/16 clock	2
11	TCLK/64 clock	3

**Table 30.10 Base clock settings in Smart card interface mode**

CCR2 setting	Base clock cycles for 1-bit period	S
BCP[2:0] setting		
0 0 0	93 clock cycles	93
0 0 1	128 clock cycles	128
0 1 0	186 clock cycles	186
0 1 1	512 clock cycles	512
1 0 0	32 clock cycles	32
1 0 1	64 clock cycles	64
1 1 0	372 clock cycles	372
1 1 1	256 clock cycles	256

[Table 30.11](#) and [Table 30.12](#) list examples of N settings in BRR in Asynchronous mode and Manchester mode. [Table 30.13](#) lists the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in Clock synchronous mode and Simple SPI mode are listed in [Table 30.16](#). Examples of BRR (N) settings in Smart card interface mode are listed in [Table 30.18](#). Examples of BRR (N) settings in Simple IIC mode are listed in [Table 30.20](#). In Smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see [section 30.7.4. Receive Data Sampling Timing and Reception Margin](#). [Table 30.14](#) and [Table 30.17](#) list the maximum bit rates with external clock input. [Table 30.15](#) lists the maximum bit rates with GPT clock input.

When either the Asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) is set to 1 in Asynchronous mode and Manchester mode, the bit rate becomes twice that listed in [Table 30.11](#) and [Table 30.12](#). When both of those registers are set to 1, the bit rate becomes four times the listed value.

**Table 30.11 Examples of BRR Settings for various Bit Rates (Asynchronous Mode and Manchester Mode)  
(1) (1 of 2)**

Bit rate (bps)	Operating Frequency TCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	-0.26	2	177	-0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0	2	129	0.16	2	155	0.16	2	159	0
300	1	207	0.16	1	255	0	2	64	0.16	2	77	0.16	2	79	0
600	1	103	0.16	1	127	0	1	129	0.16	1	155	0.16	1	159	0
1200	0	207	0.16	0	255	0	1	64	0.16	1	77	0.16	1	79	0
2400	0	103	0.16	0	127	0	0	129	0.16	0	155	0.16	0	159	0
4800	0	51	0.16	0	63	0	0	64	0.16	0	77	0.16	0	79	0
9600	0	25	0.16	0	31	0	0	32	-1.36	0	38	0.16	0	39	0
19200	0	12	0.16	0	15	0	0	15	1.73	0	19	-2.34	0	19	0
31250	0	7	0	0	9	-1.7	0	9	0	0	11	0	0	11	2.4
38400	—	—	—	0	7	0	0	7	1.73	0	9	-2.34	0	9	0

**Table 30.11 Examples of BRR Settings for various Bit Rates (Asynchronous Mode and Manchester Mode)  
(1) (2 of 2)**

Bit rate (bps)	Operating Frequency TCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	-0.17	3	70	0.03	3	75	0.48	3	79	-0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0	2	233	0.16	2	255	0
300	2	90	0.16	2	103	0.16	2	111	0	2	116	0.16	2	127	0
600	1	181	0.16	1	207	0.16	1	223	0	1	233	0.16	1	255	0
1200	1	90	0.16	1	103	0.16	1	111	0	1	116	0.16	1	127	0
2400	0	181	0.16	0	207	0.16	0	223	0	0	233	0.16	0	255	0
4800	0	90	0.16	0	103	0.16	0	111	0	0	116	0.16	0	127	0
9600	0	45	-0.93	0	51	0.16	0	55	0	0	58	-0.69	0	63	0
19200	0	22	-0.93	0	25	0.16	0	27	0	0	28	1.02	0	31	0
31250	0	13	0	0	15	0	0	16	1.2	0	17	0	0	19	-1.7
38400	—	—	—	0	12	0.16	0	13	0	0	14	-2.34	0	15	0

Note: This is an example when the CCR2.ABCS = 0, CCR2.BGDM = 0 and CCR2.ABCSE = 0.  
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.  
When both ABCS = 1 and BGDM = 1, the bit rate increases four times.

**Table 30.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode)  
(2) (1 of 3)**

Bit rate (bps)	Operating frequency TCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	-0.25	3	110	-0.02	3	132	0.13	3	145	0.33	3	177	-0.25
150	3	64	0.16	3	80	0.47	3	97	-0.35	3	106	0.39	3	129	0.16
300	2	129	0.16	2	162	-0.15	2	194	0.16	2	214	-0.07	3	64	0.16
600	2	64	0.16	2	80	0.47	2	97	-0.35	2	106	0.39	2	129	0.16
1200	1	129	0.16	1	162	-0.15	1	194	0.16	1	214	-0.07	2	64	0.16

**Table 30.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2) (2 of 3)**

Bit rate (bps)	Operating frequency TCLK (MHz)														
	20			25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
2400	1	64	0.16	1	80	0.47	1	97	-0.35	1	106	0.39	1	129	0.16
4800	0	129	0.16	0	162	-0.15	0	194	0.16	0	214	-0.07	1	64	0.16
9600	0	64	0.16	0	80	0.47	0	97	-0.35	0	106	0.39	0	129	0.16
19200	0	32	-1.36	0	40	-0.76	0	48	-0.35	0	53	-0.54	0	64	0.16
31250	0	19	0	0	24	0	0	29	0	0	32	0	0	39	0
38400	0	15	1.73	0	19	1.73	0	23	1.73	0	26	-0.54	0	32	-1.36

**Table 30.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode and Manchester Mode) (2) (3 of 3)**

Bit rate (bps)	Operating frequency TCLK (MHz)											
	50			60			100			120		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	221	-0.02	—	—	—	—	—	—	—	—	—
150	3	162	-0.15	3	194	0.16	—	—	—	—	—	—
300	3	80	0.47	3	97	-0.35	3	162	-0.15	3	194	0.16
600	2	162	-0.15	2	194	0.16	3	80	0.47	3	97	-0.35
1200	2	80	0.47	2	97	-0.35	2	162	-0.15	2	194	0.16
2400	1	162	-0.15	1	194	0.16	2	80	0.47	2	97	-0.35
4800	1	80	0.47	1	97	-0.35	1	162	-0.15	1	194	0.16
9600	0	162	-0.15	0	194	0.16	1	80	0.47	1	97	-0.35
19200	0	80	0.47	0	97	-0.35	0	162	-0.15	0	194	0.16
31250	0	49	0	0	59	0	1	24	0	0	119	0
38400	0	40	-0.76	0	48	-0.35	0	80	0.47	0	97	-0.35

Note: This is an example when the CCR2.ABCS = 0, CCR2.BGDM = 0, CCR2.ABCSE = 0 and CCR2.ABCSE2 = 0.  
 When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.  
 When both ABCS = 1 and BGDM = 1, the bit rate increases four times.

**Table 30.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester mode) (1 of 3)**

TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)	TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)
	BGDM	ABCS	ABCSE	ABCSE2			BGDM	ABCS	ABCSE	ABCSE2	
8	0	0	0	0	250000	16	0	0	0	0	500000
		1	0	0	500000			1	0	0	1000000
	1	0	0	0	1000000		1	0	0	0	2000000
		1	0	0				1	0	0	
	Don't care	Don't care	1	0	1333333		Don't care	Don't care	1	0	2666666
	Don't care	Don't care	0	1	2000000		Don't care	Don't care	0	1	4000000

**Table 30.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester mode) (2 of 3)**

TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)	TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)
	BGDM	ABCS	ABCSE	ABCSE2			BGDM	ABCS	ABCSE	ABCSE2	
9.8304	0	0	0	0	307200	17.2032	0	0	0	0	537600
		1	0	0	614400			1	0	0	1075200
	1	0	0	0	1228800		1	0	0	0	2150400
		1	0	0				1	0	0	
	Don't care	Don't care	1	0	1638400		Don't care	Don't care	1	0	2867200
	Don't care	Don't care	0	1	2457600		Don't care	Don't care	0	1	4300800
10	0	0	0	0	312500	18	0	0	0	0	562500
		1	0	0	625000			1	0	0	1125000
	1	0	0	0	1250000		1	0	0	0	2250000
		1	0	0				1	0	0	
	Don't care	Don't care	1	0	1666666		Don't care	Don't care	1	0	3000000
	Don't care	Don't care	0	1	2500000		Don't care	Don't care	0	1	4500000
12	0	0	0	0	375000	19.6608	0	0	0	0	614400
		1	0	0	750000			1	0	0	1228800
	1	0	0	0	1500000		1	0	0	0	2457600
		1	0	0				1	0	0	
	Don't care	Don't care	1	0	2000000		Don't care	Don't care	1	0	3276800
	Don't care	Don't care	0	1	3000000		Don't care	Don't care	0	1	4915200
12.288	0	0	0	0	384000	20	0	0	0	0	625000
		1	0	0	768000			1	0	0	1250000
	1	0	0	0	1536000		1	0	0	0	2500000
		1	0	0				1	0	0	
	Don't care	Don't care	1	0	2048000		Don't care	Don't care	1	0	3333333
	Don't care	Don't care	0	1	3072000		Don't care	Don't care	0	1	5000000
14	0	0	0	0	437500	25	0	0	0	0	781250
		1	0	0	875000			1	0	0	1562500
	1	0	0	0	1750000		1	0	0	0	3125000
		1	0	0				1	0	0	
	Don't care	Don't care	1	0	2333333		Don't care	Don't care	1	0	4166666
	Don't care	Don't care	0	1	3500000		Don't care	Don't care	0	1	6250000

**Table 30.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode and Manchester mode) (3 of 3)**

TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)	TCLK (MHz)	CCR2 settings				Maximum bit rate (bps)
	BGDM	ABCS	ABCSE	ABCSE2			BGDM	ABCS	ABCSE	ABCSE2	
30	0	0	0	0	937500	50	0	0	0	0	1562500
		1	0	0	1875000			1	0	0	3125000
	1	0	0	0	3750000		1	0	0	0	6250000
		1	0	0				1	0	0	
	Don't care	Don't care	1	0	5000000		Don't care	Don't care	1	0	8333333
	Don't care	Don't care	0	1	7500000		Don't care	Don't care	0	1	12500000
33	0	0	0	0	1031250	60	0	0	0	0	1875000
		1	0	0	2062500			1	0	0	3750000
	1	0	0	0	4125000		1	0	0	0	7500000
		1	0	0				1	0	0	
	Don't care	Don't care	1	0	5500000		Don't care	Don't care	1	0	10000000
	Don't care	Don't care	0	1	8250000		Don't care	Don't care	0	1	15000000
40	0	0	0	0	1250000	120	0	0	0	0	3750000
		1	0	0	2500000			1	0	0	7500000
	1	0	0	0	5000000		1	0	0	0	15000000
		1	0	0				1	0	0	
	Don't care	Don't care	1	0	6666666		Don't care	Don't care	1	0	20000000
	Don't care	Don't care	0	1	10000000		Don't care	Don't care	0	1	30000000

Note: For maximum bit rate, the CKS and BRR register values should be n = 0 and N = 0, respectively. In Manchester mode, use ABCSE = ABCSE2 = 0.

**Table 30.14 Maximum Bit Rate with External Clock Input (Asynchronous Mode)**

TCLK(MHz)	External clock (MHz)	Maximum bit rate (bps)		TCLK(MHz)	External clock (MHz)	Maximum bit rate (bps)	
		CCR2.ABCS bit = 0	CCR2.ABCS bit = 1			CCR2.ABCS bit = 0	CCR2.ABCS bit = 1
8	2	125000	250000	25	6.25	390625	781250
9.8304	2.4576	153600	307200	30	7.5	468750	937500
10	2.5	156250	312500	33	8.25	515625	1031250
12	3	187500	375000	40	10	625000	1250000
12.288	3.072	192000	384000	50	12.5	781250	1562500
14	3.5	218750	437500	60	15	937500	1875000
16	4	250000	500000	120	30	1875000	3750000
17.2032	4.3008	268800	537600				
18	4.5	281250	562500				
19.6608	4.9152	307200	614400				
20	5	312500	625000				

**Table 30.15 Maximum Bit Rate with GPT Clock Input (Asynchronous Mode)**

TCLK (MHz)	GPT clock (MHz)	Maximum bit rate (bps)		TCLK (MHz)	GPT clock (MHz)	Maximum bit rate (bps)	
		CCR2.ABCS bit = 0	CCR2.ABCS bit = 1			CCR2.ABCS bit = 0	CCR2.ABCS bit = 1
8	4	250000	500000	25	12.5	781250	1562500
9.8304	4.9152	307200	614400	30	15	937500	1875000
10	5.0	312500	625000	33	16.5	1031250	2062500
12	6	375000	750000	40	20	1250000	2500000
12.288	6.144	384000	768000	50	25	1562500	3125000
14	7.0	437500	875000	60	30	1875000	3750000
16	8	500000	1000000	120	60	3750000	7500000
17.2032	8.6016	537600	1075200				
18	9.0	562500	1125000				
19.6608	9.8304	614400	1228800				
20	10	625000	1250000				

**Table 30.16 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)**

Bit rate (bps)	Operating frequency TCLK (MHz)														
	8			10			30			60			120		
	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N	BGDM	n	N
250	0	3	124	0	3	155	—	—	—	—	—	—	—	—	—
500	0	2	249	0	3	77	0	3	233	—	—	—	—	—	—
1k	0	2	124	0	3	38	0	3	116	0	3	233	—	—	—
2.5k	0	2	49	0	1	249	0	3	46	0	3	93	0	3	187
5k	0	2	24	0	1	124	0	2	93	0	3	46	0	3	93
10k	0	1	49	0	0	249	0	2	46	0	2	93	0	3	46
25k	0	2	4	0	1	24	0	1	74	0	1	149	0	2	74
50k	0	1	9	0	0	49	0	0	149	0	1	74	0	1	149
100k	0	1	4	0	0	24	0	0	74	0	0	149	0	1	74
250k	0	1	1	0	0	9	0	0	29	0	1	14	0	1	29
500k	0	1	0	0	0	4	0	0	14	0	0	29	0	1	14
1M	0	0	1	1	0	4	1	0	14	0	0	14	0	0	29
2.5M	—	—	—	0	0	0	0	0	2	0	0	5	0	1	2
5M	—	—	—	1	0	0	1	0	2	0	0	2	0	0	5
7.5M	—	—	—	—	—	—	0	0	0	0	0	1	0	1	0
60M	—	—	—	—	—	—	—	—	—	—	—	—	1	0	0

Note: —: Can be set, but an error over 10% will occur.

**Table 30.17 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode) (1 of 2)**

TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)	TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)
8	4	4	25	12.5	12.5
10	5	5	30	15	15
12	6	6	33	16.5	16.5
14	7	7	40	20	20



**Table 30.17 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode) (2 of 2)**

TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)	TCLK (MHz)	External clock (MHz)	MAX Bit rate (Mbps)
16	8	8	50	25	25
18	9	9	60	30	30
20	10	10	120	60	60

**Table 30.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (1 of 4)**

Bit rate (bps)	Operating frequency TCLK (MHz)											
	7.1424			10.00			10.7136			13.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	0	0.00	0	1	-30	0	1	-25	0	1	-8.99

**Table 30.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (2 of 4)**

Bit rate (bps)	Operating frequency TCLK (MHz)											
	14.2848			16.00			18.00			20.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	1	0.00	0	1	12.01	0	2	-15.99	0	2	-6.66

**Table 30.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (3 of 4)**

Bit rate (bps)	Operating frequency TCLK (MHz)											
	25.00			30.00			33.00			40.00		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	3	-12.49	0	3	5.01	0	4	-7.59	0	5	-6.66

**Table 30.18 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372) (4 of 4)**

Bit rate (bps)	Operating frequency TCLK (MHz)											
	50.00			60.00			120.00					
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)			
9600	0	6	0.01	0	7	5.01	0	16	-1.17			

**Table 30.19 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)**

TCLK (MHz)	MAX Bit rate (bps)	n	N	TCLK (MHz)	MAX Bit rate (bps)	n	N
10	156250	0	0	30	468750	0	0
10.7136	167400	0	0	33	515625	0	0
13	203125	0	0	40	625000	0	0
16	250000	0	0	50	781250	0	0
18	281250	0	0	60	937500	0	0
20	312500	0	0	120	1875000	0	0
25	390625	0	0				

**Table 30.20 BRR Settings for Various Bit Rates (Simple IIC Mode) (1 of 3)**

Bit rate (bps)	Operating Frequency TCLK (MHz)											
	8			10			16			20		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10k	0	24	0	0	30	0.8	0	49	0	0	62	-0.8
25k	0	9	0	0	12	-3.8	1	4	0	0	24	0
50k	0	4	0	0	5	4.2	0	9	0	0	12	-3.8
100k	0	2	-16.7	0	2	4.2	0	4	0	0	5	4.2
250k	0	0	0	0	0	25	0	1	0	0	2	-16.7
350k										0	1	-10.7
400k										0	1	-21.9

**Table 30.20 BRR Settings for Various Bit Rates (Simple IIC Mode) (2 of 3)**

Bit rate (bps)	Operating Frequency TCLK (MHz)											
	25			30			33			40		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10k	0	77	0.2	0	93	-0.3	0	102	0.1	0	124	0
25k	0	30	0.8	0	37	-1.3	0	40	0.6	0	49	0
50k	2	0	-2.3	0	18	-1.3	0	20	-1.8	0	24	0
100k	1	1	-2.3	0	8	4.2	0	9	3.1	0	12	-3.8
250k	0	2	4.2	1	0	-6.3	1	0	3.1	0	4	0
350k	0	2	-25.6	0	2	-10.7	0	2	-1.8	1	0	-10.7
400k	0	2	-34.9	0	2	-21.9	0	2	-14.1	1	0	-21.9

**Table 30.20 BRR Settings for Various Bit Rates (Simple IIC Mode) (3 of 3)**

Bit rate (bps)	Operating Frequency TCLK (MHz)								
	50			60			120		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10k	1	38	-0.2	1	46	-0.3	1	93	-0.3
25k	0	62	-0.8	0	74	0	0	149	0
50k	0	30	0.8	0	37	-1.3	0	74	0
100k	2	0	-2.3	0	18	-1.3	0	37	-1.3
250k	0	5	4.2	1	1	-6.3	0	14	0
350k	0	4	-10.7	0	4	7.1	0	10	-2.6
400k	0	4	-21.9	0	4	-6.3	0	9	-6.3

**Table 30.21 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (1 of 4)**

Bit rate (bps)	Operating Frequency TCLK (MHz)											
	8			10			16			20		
	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)
10k	0	24	43.75 / 50.00	0	30	43.40 / 49.60	0	49	43.75 / 50.00	0	62	44.10 / 50.40
25k	0	9	17.50 / 20.00	0	12	18.20 / 20.80	1	4	17.50 / 20.00	0	24	17.50 / 20.00

**Table 30.21 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (2 of 4)**

Bit rate (bps)	Operating Frequency TCLK (MHz)											
	8			10			16			20		
	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)
50k	0	4	8.75 / 10.00	0	5	8.40 / 9.60	0	9	8.75 / 10.00	0	12	9.10 / 10.40
100k	0	2	5.25 / 6.00	0	2	4.20 / 4.80	0	4	4.37 / 5.00	0	5	4.20 / 4.80
250k	0	0	1.75 / 2.00	0	1	1.40 / 1.60	0	1	1.75 / 2.00	0	2	2.10 / 2.40
350k										0	1	1.40 / 1.60
400k										0	1	1.40 / 1.60

**Table 30.21 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (3 of 4)**

Bit rate (bps)	Operating Frequency TCLK (MHz)											
	25			30			33			40		
	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)
10k	0	77	43.68 / 49.92	0	93	43.87 / 50.13	0	102	43.70 / 49.94	0	124	43.75 / 50.00
25k	0	30	17.36 / 19.84	0	37	17.73 / 20.27	0	40	17.39 / 19.88	0	49	17.50 / 20.00
50k	2	0	8.96 / 10.24	1	4	8.87 / 10.13	0	20	8.91 / 10.18	0	24	8.75 / 10.00
100k	1	1	4.48 / 5.12	0	8	4.20 / 4.80	0	9	4.24 / 4.85	0	12	4.55 / 5.20
250k	0	2	1.68 / 1.92	1	0	1.87 / 2.13	1	0	1.70 / 1.94	0	4	1.75 / 2.00
350k	0	2	1.68 / 1.92	0	2	1.40 / 1.60	0	2	1.27 / 1.45	1	0	1.40 / 1.60
400k	0	2	1.68 / 1.92	0	2	1.40 / 1.60	0	2	1.27 / 1.45	1	0	1.40 / 1.60

**Table 30.21 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple IIC Mode) (4 of 4)**

Bit rate (bps)	Operating Frequency TCLK (MHz)								
	50			60			120		
	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)	n	N	Minimum width of High/Low Level (μs)
10k	1	38	43.68 / 49.92	1	46	43.87 / 50.03	1	93	43.87 / 50.13
25k	0	62	17.64 / 20.16	0	74	17.50 / 20.00	0	149	17.50 / 20.00
50k	0	30	8.68 / 9.92	0	37	8.87 / 10.13	0	74	8.75 / 10.00
100k	2	0	4.48 / 5.12	0	18	4.43 / 5.07	0	37	4.43 / 5.07
250k	0	5	1.68 / 1.92	1	1	1.87 / 2.13	0	14	1.75 / 2.00
350k	0	4	1.40 / 1.60	0	4	1.17 / 1.33	0	10	1.28 / 1.47
400k	0	4	1.40 / 1.60	0	4	1.17 / 1.33	0	9	1.17 / 1.33

**BRME bit (Bit Rate Modulation Enable)**

Enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

Set to 0 in Clock-synchronous mode, Simple SPI mode, Smart Card Interface mode, Manchester mode and Simple LIN mode.

Even in asynchronous mode, set this bit to 0 when ABCSE2 = 1.

**CKS[1:0] bits (Clock Select)**

The CKS[1:0] bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of the CKS[1:0] bits and the baud rate, see BRR[7:0] bits explanation.

**MDDR[7:0] bits (Modulation Duty Setting)**

When the BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of MDDR (M/256). The relationship between the MDDR setting (M) and the bit rate (B) is given in Table 30.22.

The initial value of MDDR is FFh. Bit 7 in the CCR2 is fixed to 1.

**Table 30.22 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used**

Mode*1	CCR2 settings			BRR setting	Error
	BG DM bit	AB CS bit	AB CS E bit		
Asynchronous, Multiprocessor transfer	0	0	0	$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error (\%) = \left\{ \frac{TCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1 \right\} \times 100$
	1	0	0	$N = \frac{TCLK \times 10^6}{32 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error (\%) = \left\{ \frac{TCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1 \right\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{TCLK \times 10^6}{16 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error (\%) = \left\{ \frac{TCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1 \right\} \times 100$
	x (Arbitrarily)	x (Arbitrarily)	1*2	$N = \frac{TCLK \times 10^6}{12 \times 2^{2n-1} \times (256/M) \times B} - 1$	$Error (\%) = \left\{ \frac{TCLK \times 10^6}{B \times 12 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1 \right\} \times 100$
Simple IIC*2				$N = \frac{TCLK \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	

- Note: B: Bit rate (bps)
- M: MDDR setting (128 ≤ M ≤ 255)
- N: BRR setting for bound rate generator (0 ≤ N ≤ 255)
- TCLK: Operating frequency (MHz)
- n: Determined by the settings of the CKS[1:0] as listed in Table 30.9.

- Note 1. Do not use this function in Clock-synchronous mode, Simple SPI mode, Smart card Interface mode, Manchester mode and Simple LIN mode.
- Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple IIC mode satisfy the I2C standard.

Table 30.23 and Table 30.24 list examples of N settings in BRR and M settings in MDDR in Asynchronous mode.

**Table 30.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (1 of 3)**

Bit rate (bps)	Operating frequency TCLK (MHz)														
	8					9.8304					10				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	5	236	0	0.03	0	7	(256)*1	0	0	0	10	173	1	-0.01
57600	0	3	236	0	0.03	0	4	240	0	0	0	4	236	0	0.03
115200	0	1	236	0	0.03	0	1	192	0	0	0	4	236	1	0.03
230400	0	0	236	0	0.03	0	0	192	0	0	0	1	189	1	0.14
460800	0	0	236	1	0.03	0	0	192	1	0	0	0	189	1	0.14

**Table 30.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (2 of 3)**

Bit rate (bps)	Operating frequency TCLK (MHz)														
	12					12.288					14				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	8	236	0	0.03	0	9	(256) <sup>*1</sup>	0	0	0	16	191	1	0
57600	0	5	236	0	0.03	0	4	192	0	0	0	13	236	1	0.03
115200	0	2	236	0	0.03	0	4	192	1	0	0	6	236	1	0.03
230400	0	2	236	1	0.03	0	2	230	1	-0.17	0	2	202	1	-0.11
460800	0	0	157	1	-0.18	0	0	154	1	0.26	0	0	135	1	0.14

**Table 30.23 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (1) (3 of 3)**

Bit rate (bps)	Operating frequency TCLK (MHz)														
	16					17.2032					18				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	11	236	0	0.03	0	13	(256) <sup>*1</sup>	0	0	0	18	166	1	-0.01
57600	0	7	236	0	0.03	0	6	192	0	0	0	18	249	1	-0.01
115200	0	3	236	0	0.03	0	6	192	1	0	0	8	236	1	0.03
230400	0	1	236	0	0.03	0	3	219	1	-0.2	0	1	210	0	0.14
460800	0	1	236	1	0.03	0	1	219	1	-0.2	0	0	210	0	0.14

Note: This is an example when the CCR2.ABCS = 0 and CCR2.ABCSE = 0.

Note 1. It means that the bit rate modulation function is disable. (CCR2.BRME = 0, M = 256)

**Table 30.24 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (1 of 3)**

Bit rate (bps)	Operating frequency TCLK (MHz)														
	19.6608					20					25				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	15	(256) <sup>*1</sup>	0	0	0	10	173	0	-0.01	0	11	151	0	0
57600	0	9	240	0	0	0	9	236	0	0.03	0	7	151	0	0
115200	0	4	240	0	0	0	4	236	0	0.03	0	3	151	0	0
230400	0	1	192	0	0	0	4	236	1	0.03	0	1	151	0	0
460800	0	0	192	0	0	0	0	189	0	0.14	0	0	151	0	0

**Table 30.24 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (2 of 3)**

Bit rate (bps)	Operating frequency TCLK (MHz)														
	30					33					40				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	36	194	1	0.01	0	14	143	0	0.01	0	21	173	0	-0.01
57600	0	10	173	0	-0.01	0	9	143	0	0.01	0	38	230	1	-0.01
115200	0	10	173	1	-0.01	0	4	143	0	0.01	0	9	236	0	0.03
230400	0	6	220	1	-0.09	0	4	143	1	0.01	0	4	236	0	0.03
460800	0	3	252	1	0.14	0	1	229	0	0.1	0	4	236	1	0.03

**Table 30.24 Examples of BRR and MDDR Settings for Various Bit Rates (Asynchronous Mode) (2) (3 of 3)**

Bit rate (bps)	Operating frequency TCLK (MHz)														
	50					60					120				
	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)	n	N	M	BGD M bit	Error (%)
38400	0	23	151	0	0	0	36	194	0	0.01	0	73	194	0	0.01
57600	0	15	151	0	0	0	21	173	0	-0.01	0	58	232	0	0
115200	0	7	151	0	0	0	10	173	0	-0.01	0	21	173	0	-0.01
230400	0	3	151	0	0	0	10	173	1	-0.01	0	10	173	0	-0.01
460800	0	1	151	0	0	0	6	220	1	-0.09	0	10	173	1	-0.01

Note: This is an example when the CCR2.ABCS = 0 and CCR2.ABCSE = 0.

Note 1. It means that the bit rate modulation function is disable. (CCR2.BRME = 0, M = 256)

### 30.2.8 CCR3 : Common Control Register 3

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	BLK	GM	—	ACSO	CKE[1:0]	—	—	DEN	FM	MP	MOD[2:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXDE SEL	STP	SINV	LSBF	—	—	CHR[1:0]	BPEN	—	—	—	—	—	—	CPOL	CPHA
Value after reset:	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	CPHA	Clock Phase Select Valid in Clock synchronous mode and Simple SPI mode. Set this bit only when CCR0.TE = 0 and RE = 0. 0: Data is sampled at an odd edge and changes at an even edge (clock is delayed) 1: Data changes at an odd edge and is sampled at an even edge (clock is not delayed)	R/W
1	CPOL	Clock Polarity Select Valid in Clock synchronous mode and Simple SPI mode. Set this bit only when CCR0.TE = 0 and RE = 0. 0: SCKn in idle state is 0 1: SCKn in idle state is 1	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	BPEN	Synchronizer Bypass Enable This bit controls whether to bypass the synchronizer circuit between the bus clock and operation clock. 0: Synchronizer circuit is not bypassed 1: Synchronizer circuit is bypassed	R/W
9:8	CHR[1:0]	Character Length Valid in Asynchronous mode and Manchester mode*1 Select the data length for transmission and reception. 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*2	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12	LSBF	LSB First select Set this bit to 0 in Simple IIC mode. Set this bit to 1 in Simple LIN mode. 0: MSB-first 1: LSB-first	R/W
13	SINV	Transmitted/Received Data Invert Set this bit to 0 in Simple IIC mode. The level of communication pins (RXDn/TXDn) are controlled by combination of this bit and CCR1.TINV/RINV. For details, see <a href="#">Figure 30.3</a> . 0: TDR contents are transmitted to TSR as is. RSR contents are stored to RDR as is. 1: TDR contents are inverted before being transmitted to TSR. RSR contents are inverted and stored to RDR.	R/W
14	STP	Stop Bit Length Valid in Asynchronous mode, Manchester mode, and Simple LIN mode 0: 1 stop bit/break delimiter length is 1-bit 1: 2 stop bits/break delimiter length is 2-bit	R/W
15	RXDESEL	Asynchronous Start Bit Edge Detection Select Valid only in Asynchronous mode Set this bit to 1 in Simple LIN mode. 0: The low level on the RXDn pin is detected as the start bit 1: A falling edge on the RXDn pin is detected as the start bit	R/W
18:16	MOD[2:0]	Communication Mode Select Select the SCI communication mode. 0 0 0: Asynchronous mode (multi-processor mode) 0 0 1: Smart card interface mode 0 1 0: Clock synchronous mode 0 1 1: Simple SPI mode 1 0 0: Simple IIC mode 1 0 1: Manchester mode 1 1 0: Simple LIN mode 1 1 1: Setting prohibited	R/W
19	MP	Multi-Processor Mode Valid in Asynchronous mode and Manchester mode 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W
20	FM	FIFO Mode Select Valid in Asynchronous mode (including Multi-processor mode), Clock synchronous mode, and Simple SPI mode 0: TDR register and RDR register are non-FIFO buffer configuration 1: TDR register and RDR register are FIFO buffer configuration	R/W
21	DEN	Driver Enable 0: RS-485 driver control function disabled 1: RS-485 driver control function enabled	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
25:24	CKE[1:0]	Clock Enable See <a href="#">Table 30.26</a> for details.	R/W
26	ACS0	Asynchronous Mode Clock Source Select Valid only in Asynchronous mode. These bits for SCI channels other than SCI1 are reserved. 0: External clock input 1: Logical AND of compare matches output from the internal GPT	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	GM	GSM Mode Valid only in Smart card interface mode. 0: Non-GSM mode operation 1: GSM mode operation	R/W

Bit	Symbol	Function	R/W
29	BLK	Block Transfer Mode Valid only in Smart card interface mode. 0: Non-block transfer mode operation 1: Block transfer mode operation	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. In other than Asynchronous mode and Manchester mode, this bit setting is invalid and a fixed data length of 8 bits is used. In the Simple LIN mode, only the data length of 8 bits can be used, so set it to the initial value.

Note 2. LSB first is fixed and the MSB (bit 7) in TDR is not transmitted in transmission.

Note 3. GPT clock is selectable for SCI1.

### CPHA bit (Clock Phase Select)

The CPHA bit selects the phase of the clock signal output through the SCKn pin. For details, see [Figure 30.99](#).

Set the bit to 1 in other than Simple SPI mode and Clock synchronous mode.

### CPOL bit (Clock Polarity Select)

The CPOL bit selects the polarity of the clock signal output through the SCKn pin. For details, see [Figure 30.99](#).

Set the bit to 1 in other than Simple SPI mode and Clock synchronous mode.

### BPEN bit (Synchronizer Bypass Enable)

The BPEN bit selects whether to enable or disable the synchronization bypass function. Set this bit to 1 when the bus clock (PCLK) is also used as operation clock (TCLK). For details, see [Figure 30.2](#) and [section 30.18. Synchronizer Bypass Function](#).

Note: For details of this bit setting, see [section 30.19.17. Notes on CCR3.BPEN bit setting](#).

### CHR[1:0] bits (Character Length)

Selects the data length for transmission and reception.

Except for Asynchronous mode and Manchester mode, a fixed data length of 8 bits is used.

### LSBF bit (LSB First select)

Select whether to transmit/receive data in MSB-first or LSB-first.

### SINV bit (Transmitted/Received Data Invert)

SINV can invert the transmit data bit from TDR to TSR, and can invert the received data from RSR to RDR. The SINV bit does not affect the logic level of the parity bit. To invert the parity bit, invert the CCR1.PM.

### STP bit (Stop Bit Length)

Selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of the STP bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

In addition, it is used as the break delimiter length setting when sending start frame in Simple LIN mode.

### RXDESEL bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in Asynchronous mode. When a break occurs, data receiving operation depends on the settings of the RXDESEL bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.

Set this bit to 1 in Simple LIN mode. Set this bit to 0 in modes except of Asynchronous mode and Simple LIN mode.

### MOD[2:0] bits (Communication Mode Select)

Selects the SCI communication mode.



**Table 30.25 Relationship between communication mode selection bits (MOD[2:0]) and other operation mode setting bits**

Communication mode	Asynchronous				Smart card interface	Clock synchronous	Simple SPI				Simple IIC	Manchester	Simple LIN
CCR3.MOD[2:0]	000b				001b	010b	011b				100b	101b	110b
CCR3.MP	0		1		—	—	—				—	0   1	—
CCR3.FM	0	1	0	1	—	0   1	0	1	—	—	—	—	
CCR3.DEN	0   1	0   1	0   1	0   1	—	—	—				—	—	
CCR0.SSE	—				—	—	0	1	0	1	—	—	—

Note: — is prohibited setting.

### MP bit (Multi-Processor Mode)

Disables/enables multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

### FM bit (FIFO Mode Select)

When the FM bit is set to 1, the TDR register / RDR register switches to FIFO configuration, and transmit FIFO (TDR register) / receive FIFO (RDR register) can be used for serial transmission/reception.

### DEN bit (Driver Enable)

Select RS-485 driver control function disabled or enabled.

### CKE[1:0] bits (Clock Enable)

**Table 30.26 Association between Clock Enable bits CKE[1:0] and operation mode (1 of 2)**

CKE[1:0]	Function
00b	In Asynchronous mode: On-chip baud rate generator The SCKn pin is available for use as an I/O port in accord with the I/O port settings. In Manchester mode and Simple LIN mode: On-chip baud rate generator The SCKn pin functions as I/O port. In Clock synchronous mode and Simple SPI mode: Internal clock (master operation) The SCKn pin functions as the clock output pin. In Smart card interface mode and when CCR3.GM = 0: Output disabled (the SCKn pin is available for use as an I/O port in accord with the I/O port settings). In Smart card interface mode and when CCR3.GM = 1: Output fixed low
01b	In Asynchronous mode: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. In Manchester mode and Simple LIN mode: Prohibited In Clock synchronous mode and Simple SPI mode: Internal clock (master operation) The SCKn pin functions as the clock output pin. In Smart card interface mode and when CCR3.GM = 0: Clock output In Smart card interface mode and when CCR3.GM = 1: Clock output

**Table 30.26 Association between Clock Enable bits CKE[1:0] and operation mode (2 of 2)**

CKE[1:0]	Function
10b	<p>In Asynchronous mode: External clock or GPT clock</p> <ul style="list-style-type: none"> <li>When using the external clock 16 times the bit rate should be input from the SCKn pin when CCR2.ABCS bit is 0. Input a clock signal with a frequency 8 times the bit rate when the CCR2.ABCS bit is 1.</li> <li>When using the GPT clock<sup>*3</sup> The SCKn pin is available for use as an I/O port in accord with the I/O port settings when the GPT clock is used.</li> </ul> <p>In Manchester mode and Simple LIN mode: Prohibited</p> <p>In Clock synchronous mode and Simple SPI mode: External clock (slave operation) The SCKn pin functions as the clock input pin.</p> <p>In Smart card interface mode and when CCR3.GM = 0: Prohibited</p> <p>In Smart card interface mode and when CCR3.GM = 1: Output fixed high</p>
11b	<p>In Asynchronous mode: External clock or GPT clock</p> <ul style="list-style-type: none"> <li>When using the external clock 16 times the bit rate should be input from the SCKn pin when CCR2.ABCS bit is 0. Input a clock signal with a frequency 8 times the bit rate when the CCR2.ABCS bit is 1.</li> <li>When using the GPT clock<sup>*3</sup> The SCKn pin is available for use as an I/O port in accord with the I/O port settings when the GPT clock is used.</li> </ul> <p>In Manchester mode and Simple LIN mode: Prohibited</p> <p>In Clock synchronous mode and Simple SPI mode: External clock (slave operation) The SCKn pin functions as the clock input pin.</p> <p>In Smart card interface mode and when CCR3.GM = 0: Prohibited</p> <p>In Smart card interface mode and when CCR3.GM = 1: Clock output</p>

The CKE[1:0] bits select the clock source and SCKn pin function.

The combination of these bit settings and of the CCR3.ACS0 bit sets the internal GPT clock.

In Smart card interface mode, these bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, see [section 30.7.8. Clock Output Control](#).

### GM bit (GSM Mode)

Setting the GM bit to 1 allows GSM mode operation.

In GSM mode, the CSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, see [section 30.7.6. Serial Data Transmission \(Except in Block Transfer Mode\)](#), [section 30.7.8. Clock Output Control](#).

### ACS0 bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the Asynchronous mode.

The ACS0 bit is valid in Asynchronous mode (CCR3.MOD[2:0] = 000b) and when an external clock input is selected (CCR3.CKE[1:0] = 10b or 11b). This bit is used to select an external clock input or the logical AND of compare matches output from the internal GPT.

Set the ACS0 bit to 0 in other than Asynchronous mode.

For SCI1, the GTIOCmA output (m = 3, 4) of GPT can be set as the serial transfer base clock. See [Table 30.27](#) for details.

These bits for SCI channels other than SCI1 and SCI2 are reserved. The write values to these bits for other than SCI1 and SCI2 should be 0.

**Table 30.27 Association between SCI channels and compare match outputs**

SCI	GPT	Compare match output
SCI1	GPT3	GTIOC3A
	GPT4	GTIOC4A

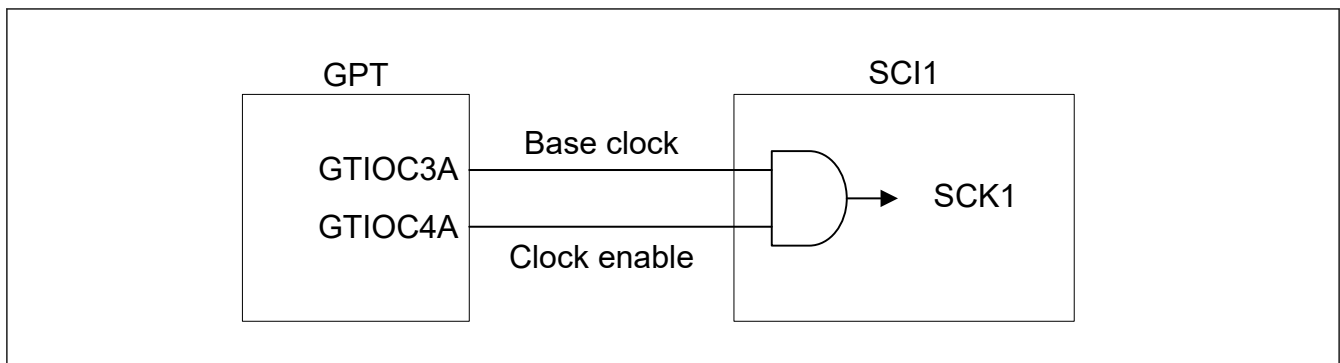
Figure 30.4, Figure 30.5 show a setting example of when GTIOC3A and GTIOC4A are selected for output. This figure shows an example when GPT clock is input to SCI1.

When generating 187.5 kbps of GPT average transfer rate for PCLKD = 32 MHz:

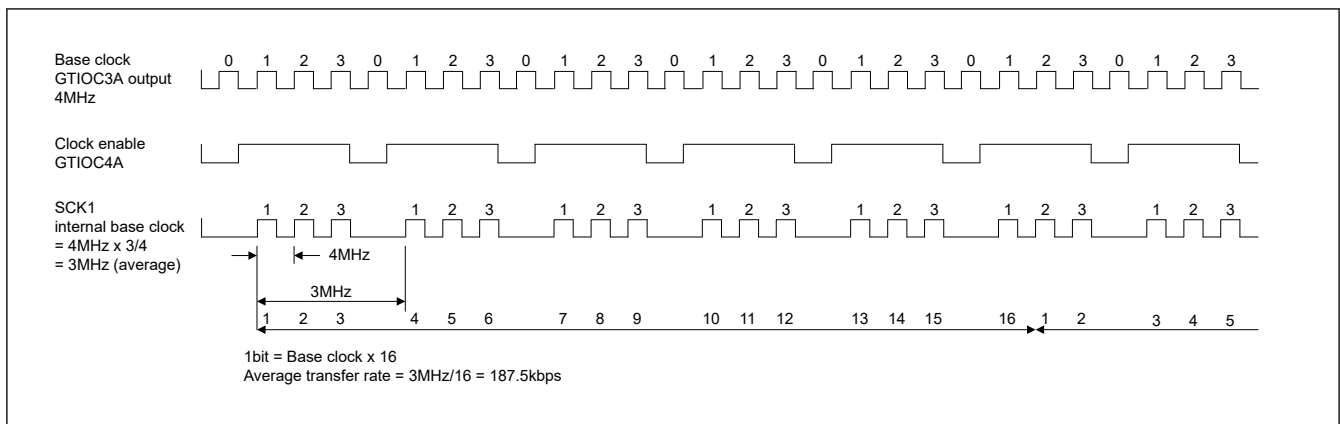
1. Generate a frequency of 4 MHz using GTIOC3A as the base clock.
2. Generate 3/4 clock enable of base clock to set an average transfer rate of 3 MHz/16 = 187.5 kbps using GTIOC4A.

Setting examples of GPT and SCI

- GPT3.GTSSR = 0x80000000, GPT4.GTSSR = 0x80000000 (enable software counter start)
- GPT3.GTPR = 0x00000007, GPT4.GTPR = 0x0000001F (maximum count value of GTCNT)
- GPT3.GTCCRA = 0x00000003, GPT4.GTCCRA = 0x00000007 (compare match value)
- GPT3.GTCR = 0x00000000, GPT4.GTCR = 0x00000000 (Saw-wave PWM mode, timer prescaler is PCLKD)
- GPT3.GTIOR = 0x00000306, GPT4.GTIOR = 0x00000306 (initial output low, output high at GTCCRA compare match, output low at cycle end)
- CCR3.CKE[1:0] = 10b (external clock input or GPT clock input is selected)
- CCR3.ACS0 = 1, CCR2.ABCSE = 1, CCR2.ABCS=0 (GPT clock input is selected with 16 base clock cycles for 1-bit period)
- GPT3.GTSTR = 0x00000018 (software start GTCNT counter)



**Figure 30.4 Example of GPT and SCI connection**



**Figure 30.5 Example of average transfer rate setting when GPT clock is input**

**BLK bit (Block Transfer Mode)**

Setting the BLK bit to 1 allows block transfer mode operation.

For details, see [section 30.7.3. Block Transfer Mode](#).

**30.2.9 CCR4 : Common Control Register 4**

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AET	ATT[2:0]			AJD	AST[2:0]			—	—	—	—	SCKSEL	—	ATEN	ASEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CMPD[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	CMPD[8:0]	Compare Match Data Valid only in Asynchronous mode.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ASEN	Adjust Receive Sampling Timing Enable Valid in Asynchronous mode using internal clock, Simple LIN mode using internal clock, Clock-synchronous mode operating as master, and Simple SPI mode operating as master. 0: Adjust sampling timing disabled 1: Adjust sampling timing enabled	R/W
17	ATEN	Adjust Transmit Timing Enable Valid only in Asynchronous mode using internal clock. 0: Adjust transmit timing disabled 1: Adjust transmit timing enabled	R/W
18	—	This bit is read as 0. The write value should be 0.	R/W
19	SCKSEL	Master receive clock selection Valid only in Clock synchronous and Simple-SPI mode. 0: Master receive clock is disabled 1: Master receive clock is enabled	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
26:24	AST[2:0]	Adjustment Value for Receive Sampling Timing This bit enables only when ASEN is 1. In Asynchronous mode and Simple LIN mode using internal clock: The sampling timing of RXDn pin is adjusted from the middle of bit by the following formula. Adjustment sampling timing = base clock × the setting value of AST[2:0]. In Clock synchronous mode and Simple SPI mode using internal clock: The RXDn sampling timing can be adjusted by delaying 1 to 4 TCLK 0 0 0: 1-TCLK delay 0 0 1: 2-TCLK delay 0 1 0: 3-TCLK delay 0 1 1: 4-TCLK delay Others: Setting prohibited	R/W
27	AJD	Adjustment Direction for Receive Sampling Timing Valid in Asynchronous mode using internal clock and Simple LIN mode using internal clock. This bit enables only when ASEN is 1. Adjustment direction for RXDn receive sampling timing is determined by this bit. For details, see <a href="#">section 30.3.10. The function of adjust receive sampling timing (Asynchronous Mode)</a> . 0: The sampling timing is adjusted backward to the middle of bit. 1: The sampling timing is adjusted forward to the middle of bit.	R/W

Bit	Symbol	Function	R/W
30:28	ATT[2:0]	Adjustment Value for Transmit Timing Valid in Asynchronous mode using internal clock and Simple LIN mode using internal clock. This bit enables only when ATEN is 1. The selected edge timing of TXDn is adjusted by the following formula. Adjustment edge timing = base clock × the setting value of ATT[2:0] This setting timing is limited by setting the base clock cycles. For details, see <a href="#">section 30.3.11. The function of adjust transmit timing (Asynchronous Mode)</a> .	R/W
31	AET	Adjustment Edge for Transmit Timing Valid in Asynchronous mode using internal clock and Simple LIN mode using internal clock. The adjustable edge is set by this bit. This bit enables only when ATEN is 1. For details, see <a href="#">section 30.3.11. The function of adjust transmit timing (Asynchronous Mode)</a> .  0: When CCR1.TINV is 0, adjust the rising edge timing. When CCR1.TINV is 1, adjust the falling edge timing. 1: When CCR1.TINV is 0, adjust the falling edge timing. When CCR1.TINV is 1, adjust the rising edge timing.	R/W

Note: S-TYPE-3, P-TYPE-3

### CMPD[8:0] bits (Compare Match Data)

Set the comparison data for receive data when address match function is enabled (CCR0.DCME = 1). CCR4.CMPD[8:0] should be written while CCR0.DCME is 0.

For the comparison data, it can select length from three types, they are CMPD[6:0] with 7-bit length, CMPD[7:0] with 8-bit length, and CMPD[8:0] with 9-bit length.

Note: If the description in this document and the timing chart do not specify the ASEN / ATEN setting value, it means that the reception sampling adjustment function/transmission timing adjustment function are OFF (CCR4.ASEN = 0, CCR4.ATEN = 0).

### ASEN bit (Adjust Receive Sampling Timing Enable)

When the ASEN bit is 1, the receive sampling timing adjustment function is enabled. Control is different in Asynchronous mode, Simple LIN mode, Clock-synchronous mode, and Simple SPI mode.

For details of using internal clock in Asynchronous mode, see [section 30.6.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used](#). The operation when the Simple LIN mode internal clock is selected is the same as when the Asynchronous mode internal clock is selected.

For details of Clock synchronous mode operating as master and Simple SPI mode operating as master, see [section 30.6.6. Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode](#). Only the digital delay of the master mode receive sampling clock can be controlled by this bit.

### ATEN bit (Adjust Transmit Timing Enable)

When the ATEN bit is 1, the transmission timing adjustment function is enabled. The transmission timing adjustment function can adjust the edge timing of the waveform output from the TXDn pin. For details, see [section 30.3.11. The function of adjust transmit timing \(Asynchronous Mode\)](#).

### SCKSEL bit (Master receive clock selection)

In Clock synchronous or Simple-SPI mode, SCKSEL bit selects whether to enable or disable the internal clock for master reception.

### AST[2:0] bits (Adjustment Value for Receive Sampling Timing)

When the ASEN bit is 1, the receive sampling timing can be adjusted according to this bit setting value.

In Asynchronous mode and Simple LIN mode using internal clock.

The sampling timing of RXDn pin is adjusted from the middle of bit by the following formula. This setting value is limited by setting the base clock cycles. For details, see [section 30.3.10. The function of adjust receive sampling timing \(Asynchronous Mode\)](#).

Adjustment sampling timing = base clock × the setting value of AST[2:0].

In Clock-synchronous mode and Simple SPI mode using internal clock

The sampling timing of RXDn pin can be adjusted by delaying 1 to 4 TCLK. For details, see [section 30.6.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used.](#)

000: 1-TCLK delay

001: 2-TCLK delay

010: 3-TCLK delay

011: 4-TCLK delay

1xx: Setting prohibited

Note: In Clock-synchronous mode and Simple SPI mode, it is recommended to set delay time within half the period of the SCK clock cycle.

**AJD bit (Adjustment Direction for Receive Sampling Timing)**

Set the RXDn pin sampling timing adjustment direction from the bit center to the rear or front. For details, see [section 30.3.10. The function of adjust receive sampling timing \(Asynchronous Mode\).](#)

**ATT[2:0] bits (Adjustment Value for Transmit Timing)**

The edge timing of the TXDn pin specified by the AET bit is adjusted by the base clock × ATT[2:0] setting value. The upper limit of the adjustment time that can be set is limited by the number of base clock cycles. For details, see [section 30.3.11. The function of adjust transmit timing \(Asynchronous Mode\).](#)

**AET bit (Adjustment Edge for Transmit Timing)**

Set the TXDn pin edge for timing adjustment. For details, see [section 30.3.11. The function of adjust transmit timing \(Asynchronous Mode\).](#)

**30.2.10 ICR : Simple IIC Control Register**

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	IICSCLS[1:0]		IICSDAS[1:0]		—	IICST PREQ	IICRS TARE Q	IICSTA REQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IICAC KT	—	—	—	IICCS C	IICINT M	—	—	—	IICDL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	IICDL[4:0]	SDA Delay Output Select The following cycles are of the clock signal from the on-chip baud rate generator: 0x00: No output delay 0x01: 0 to 1 cycle 0x02: 1 to 2 cycles 0x03: 2 to 3 cycles 0x04: 3 to 4 cycles 0x05: 4 to 5 cycles : 0x1E: 29 to 30 cycles 0x1F: 30 to 31 cycles	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	IICINTM	IIC Interrupt Mode Select 0: Use ACK/NACK interrupts 1: Use reception and transmission interrupts	R/W

Bit	Symbol	Function	R/W
9	IICSC	Clock Synchronization 0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
13	IICACKT	ACK Transmission Data 0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	IICSTAREQ	Start Condition Generation 0: A start condition is not generated 1: A start condition is generated *1 *3 *4 *5	R/W
17	IICRSTAREQ	Restart Condition Generation 0: A restart condition is not generated 1: A restart condition is generated *2 *3 *4 *5	R/W
18	IICSTPREQ	Stop Condition Generation 0: A stop condition is not generated. 1: A stop condition is generated *2 *3 *4 *5	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
21:20	IICSDAS[1:0]	SDA Output Select 0 0: Serial data output 0 1: Generate a start, restart, or stop condition 1 0: Output the low level on the SDA pin 1 1: Place the SDA pin in the high-impedance state	R/W
23:22	IICSCLS[1:0]	SCL Output Select 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition 1 0: Output the low level on the SCL pin 1 1: Place the SCL pin in the high-impedance state	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. In bus free state, perform the start condition generation.

Note 2. In bus busy state, perform restart or stop condition generation when the SCLn pin after acknowledgment described in [Figure 30.86](#) and [Figure 30.87](#) is low level.

Note 3. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.

Note 4. Execute the generation of a condition after the value of the IICSTIF flag is 0.

Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

### IICDL[4:0] bits (SDA Delay Output Select)

The IICDL[4:0] bits are used to set a delay for output on the SDA pin relative to the falling edge of the output on the SCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generators the base. The signal obtained by frequency-dividing TCLK by the divisor set in CCR2.CKS[1:0] is supplied as the clock signal from the on-chip baud rate generator.

Set these bits to 00000b unless operation is in simple IIC mode. In simple IIC mode, set the bits to a value in the range from 00001b to 11111b.

### IICINTM bit (IIC Interrupt Mode Select)

The IICINTM bit selects the sources of interrupt requests in Simple IIC mode.

### IICSC bit (Clock Synchronization)

Set the IICSC bit to 1 if the internally generated SCLn clock signal is to be synchronized when the SCLn pin is placed at the low level and when a wait is inserted by other device, for example.

The SCLn clock signal is not synchronized if the IICSC bit is 0. The SCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SCLn pin.

Set the IICSC bit to 1 except during debugging.

**IICACKT bit (ACK Transmission Data)**

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

**IICSTAREQ bit (Start Condition Generation)**

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.

To generate the start condition after generating the stop condition, start the generation of the start condition with a half cycle period of the bit rate from the stop condition generation interrupt (STI) request output.

[ Setting condition ]

- Writing 1 to the bit

[ Clearing condition ]

- Completion of generation of the start condition

**IICRSTAREQ bit (Restart Condition Generation)**

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.

[ Setting condition ]

- Writing 1 to the bit

[ Clearing condition ]

- Completion of generation of the restart condition

**IICSTPREQ bit (Stop Condition Generation)**

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[ Setting condition ]

- Writing 1 to the bit

[ Clearing condition ]

- Completion of generation of the stop condition

**IICSDAS[1:0] bits (SDA Output Select)**

The IICSDAS[1:0] bits control output from the SDA<sub>n</sub> pin.

**IICSCLS[1:0] bits (SCL Output Select)**

The IICSCLS[1:0] bits control output from the SCL<sub>n</sub> pin.

**30.2.11 FCR : FIFO Control Register**

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Bit field:	—	—	—	RSTRG[4:0]				RFRS T	—	—	RTRG[4:0]						
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit field:	TFRS T	—	—	TTRG[4:0]				—	—	—	—	—	—	—	—	—	DRES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	



Bit	Symbol	Function	R/W
0	DRES	Receive Data Ready Error Select Bit Valid in Asynchronous mode. This bit selects the interrupt request for a reception data ready detection. 0: Reception data full interrupt (SCIn_RXI) 1: Receive error interrupt (SCIn_ERI)	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
12:8	TTRG[4:0]	Transmit FIFO Data Trigger Number Valid in Asynchronous mode (including Multiprocessor mode), Clock synchronous mode, and Simple SPI mode. Trigger number must be set to 15 or less. 0x00: Trigger number 0 ⋮ 0x1F: Trigger number 31	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	TFRST	Transmit FIFO Data Register Reset This bit enables only when CCR3.FM is 1. The read value is always 0. 0: When set, this bit is invalid and does not affect operation 1: The number of data stored in Transmit-FIFO (TDR register) is 0	W
20:16	RTRG[4:0]	Receive FIFO Data Trigger Number Valid in Asynchronous mode (including Multi processor mode), Clock synchronous mode, and Simple SPI mode. Trigger number must be set to 15 or less. 0x00: Trigger number 0 ⋮ 0x1F: Trigger number 31	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	RFRST	Receive FIFO Data Register Reset This bit enables only when CCR3.FM is 1. The read value is always 0. 0: When set, this bit is invalid and does not affect operation 1: The number of data stored in Receive-FIFO(RDR register) is 0	W
28:24	RSTRG[4:0]	RTS Output Active Trigger Number Select Valid in Asynchronous mode (including Multi processor mode) and Clock synchronous mode. This bit enables only when CCR3.FM = 1 and CCR1.CTSE = 0 and CCR0.SSE = 0. Trigger number must be set to 15 or less. 0x00: Trigger number 0 ⋮ 0x1F: Trigger number 31	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

### DRES bit (Receive Data Ready Error Select Bit)

Select whether the detection of receive data ready (FRSR.DR flag = 1) is the cause of SCIn\_RXI interrupt request or the cause of SCIn\_ERI interrupt request.

### TTRG[4:0] bits (Transmit FIFO Data Trigger Number)

The TDRE flag is set to 1 when the quantity of transmit data in the transmit-FIFO(TDR register) is equal to or less than the specified transmit triggering number. If CCR0.TIE = 1, SCIn\_TXI interrupt request occurs.

Note: Trigger number must be set to 15. When the trigger number is set to 16 or more, unexpected SCIn\_TXI interrupt occurs.

### TFRST bit (Transmit FIFO Data Register Reset)

When the TFRST bit is set to 1, the number of the transmission data stored in Transmit-FIFO(TDR register) is 0.

**RTRG[4:0] bits (Receive FIFO Data Trigger Number)**

The CSR.RDRF flag is set to 1 when the quantity of receive data in the receive-FIFO(RDR register) is equal to or greater than the specified receive triggering number. If CCR0.RIE = 1, SCIn\_RXI interrupt request occurs. When FCR.RTRG is set to 0, RDRF bit is set if the quantity of data in receive-FIFO is greater than or equal to 1.

Note: Trigger number must be set to 15. When the trigger number is set to 16 or more, unexpected SCIn\_RXI interrupt occurs.

**RFRST bit (Receive FIFO Data Register Reset)**

When the RFRST bit is set to 1, the number of the reception data stored in Receive-FIFO(RDR register) is 0.

**RSTRG[4:0] bits (RTS Output Active Trigger Number Select)**

When the quantity of receive data stored in the receive-FIFO (RDR register) is equal to or greater than this number, the RTSn signal is in the High state. When FCR.RSTRG is set to 0, RTSn is in the high state if the quantity of data in receive FIFO is greater than or equal to 1.

Note: Trigger number must be set to 15. When the trigger number is set to 16 or more, RTSn goes to a high state at unexpected timing.

**30.2.12 MCR : Manchester Control Register**

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0)  
SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SBER EN	SYER EN	PFER EN	—	—	RPPAT[1:0]	RPLEN[3:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TPPAT[1:0]	TPLEN[3:0]				—	SBSE L	SYNS EL	SYNV AL	—	ERTE N	TMPO L	RMPO L	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RMPOL	Polarity of Received Manchester Code Sets the polarity of the received Manchester code. 0: Logic 0 is coded as a zero-to-one transition in Manchester code Logic 1 is coded as a one-to-zero transition in Manchester code 1: Logic 0 is coded as a one-to-zero transition in Manchester code Logic 1 is coded as a zero-to-one transition in Manchester code	R/W
1	TMPOL	Polarity of Transmit Manchester Code Sets the polarity of the transmit Manchester code 0: Logic 0 is coded as a zero-to-one transition in Manchester code Logic 1 is coded as a one-to-zero transition in Manchester code 1: Logic 0 is coded as a one-to-zero transition in Manchester code Logic 1 is coded as a zero-to-one transition in Manchester code	R/W
2	ERTEN	Manchester Edge Retiming Enable Sets the receive retiming function. 0: Disables the receive retiming function 1: Enables the receive retiming function	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	SYNVAL	<p>SYNC value Setting Sets the SYNC type of the start bit(s) in the Manchester code. When the start bit area consists of one bit (SBSEL = 0)</p> <ul style="list-style-type: none"> <li>• When transmitting: <ul style="list-style-type: none"> <li>0: The start bit is added as a zero-to-one transition</li> <li>1: The start bit is added as a one-to-zero transition</li> </ul> </li> <li>• When receiving: <ul style="list-style-type: none"> <li>0: Only when the start bit is a zero-to-one transition, the data is received. Other cases are determined as an error.</li> <li>1: Only when the start bit is a one-to-zero transition, the data is received. Other cases are determined as an error.</li> </ul> </li> </ul> <p>When the start bit area consists of three bits (SBSEL = 1)</p> <ul style="list-style-type: none"> <li>• When transmitting: <ul style="list-style-type: none"> <li>0: The start bits are added as a zero-to-one transition (DATA SYNC)</li> <li>1: The start bits are coded as a one-to-zero transition (COMMAND SYNC)</li> </ul> </li> <li>• When receiving: <ul style="list-style-type: none"> <li>When the start bit area consists of three bits, data is received regardless of the value of this bit.</li> </ul> </li> </ul>	R/W
5	SYNSEL	<p>SYNC Select</p> <ul style="list-style-type: none"> <li>0: The start bit pattern is set with the SYNVAL bit</li> <li>1: The start bit pattern is set with the TSYNC bit</li> </ul>	R/W
6	SBSEL	<p>Start Bit Select</p> <ul style="list-style-type: none"> <li>0: The start bit area consists of one bit</li> <li>1: The start bit area consists of three bits (COMMAND SYNC or DATA SYNC)</li> </ul>	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
11:8	TPLEN[3:0]	<p>Transmit Preface Length Set the preface length of the transmit data in Manchester mode.</p> <ul style="list-style-type: none"> <li>0x0: Disables the transmit preface generation</li> <li>Others: Transmit preface length (bit length)</li> </ul>	R/W
13:12	TPPAT[1:0]	<p>Transmit Preface Pattern Set the preface pattern of the transmit data.</p> <ul style="list-style-type: none"> <li>0 0: ALL ZERO</li> <li>0 1: ZERO ONE</li> <li>1 0: ONE ZERO</li> <li>1 1: ALL ONE</li> </ul>	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
19:16	RPLEN[3:0]	<p>Receive Preface Length Set the preface length in received frames when Manchester mode is enabled.</p> <ul style="list-style-type: none"> <li>0x0: Disables the receive preface generation</li> <li>Others: Receive preface length (bit length)</li> </ul>	R/W
21:20	RPPAT[1:0]	<p>Receive Preface Pattern Set the preface pattern of received frames</p> <ul style="list-style-type: none"> <li>0 0: ALL ZERO</li> <li>0 1: ZERO ONE</li> <li>1 0: ONE ZERO</li> <li>1 1: ALL ONE</li> </ul>	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	PFEREN	<p>Preface Error Enable Specifies whether to handle a preface error as an interrupt source.</p> <ul style="list-style-type: none"> <li>0: Does not handle a preface error as an interrupt source</li> <li>1: Handles a preface error as an interrupt source</li> </ul>	R/W
25	SYEREN	<p>Receive SYNC Error Enable Specifies whether to handle a receive SYNC error as an interrupt source.</p> <ul style="list-style-type: none"> <li>0: Does not handle a receive SYNC error as an interrupt source</li> <li>1: Handles a receive SYNC error as an interrupt source</li> </ul>	R/W
26	SBEREN	<p>Start Bit Error Enable Specifies whether to handle a start bit error as an interrupt source.</p> <ul style="list-style-type: none"> <li>0: Does not handle a start bit error as an interrupt source</li> <li>1: Handles a start bit error as an interrupt source</li> </ul>	R/W

Bit	Symbol	Function	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

#### **RMPOLE bit (Polarity of Received Manchester Code)**

The RMPOLE bit sets the polarity of the received Manchester code. For details, see [section 30.5.7. Serial Data Reception in Manchester Mode](#).

#### **TMPOLE bit (Polarity of Transmit Manchester Code)**

The TMPOLE bit sets the polarity of the transmit Manchester code. For details, see [section 30.5.6. Serial data transmission in Manchester mode](#).

#### **ERTEN bit (Manchester Edge Retiming Enable)**

The ERTEN bit sets the receive retiming function in Manchester mode.

For information on the receive retiming function, see [section 30.5.9. Receive Retiming](#).

#### **SYNVAL bit (SYNC value Setting)**

The SYNVAL bit is valid when the SYNSEL bit of the MCR is set to 0.

The SYNC type can be set by combining this bit and the SBSEL bit.

For the start bit area determined by the combination of this bit and the SBSEL bit, see [Figure 30.54](#) and [Figure 30.55](#).

#### **SYNSEL bit (SYNC Select)**

The SYNSEL bit is valid when the SBSEL bit of the MCR is set to 1. This bit determines the destination to be referred to for setting the SYNC type of the start bit area added to Manchester frames.

When this bit is set to 0, the SYNVAL bit of the MCR is referred to.

When this bit is set to 1, the TSYNC bit in the TDR register is referred to.

#### **SBSEL bit (Start Bit Select)**

The SBSEL bit sets the start bit area in Manchester frames.

When this bit is set to 1, the start bit area added to each frame consists of three bits, and the SYNSEL and SYNVAL bits in this register are valid.

When this bit is set to 0, the start bit area added to each frame consists of one bit.

#### **TPLEN[3:0] bits (Transmit Preface Length)**

The TPLEN[3:0] bits set the preface bit length of the transmit data in Manchester mode.

The settable range is 0x0 to 0xF (0d to 15d). The setting of 0x0 disables the transmit preface, which is not added.

#### **TPPAT[1:0] bits (Transmit Preface Pattern)**

The TPPAT[1:0] bits set one of the four preface patterns in Manchester mode. For the transmit data when the TPPAT[1:0] bits are set, see [Figure 30.53](#).

When these bits are set to 00b, the preface area is set to all 0s.

When these bits are set to 01b, the preface area is set to the zero-one-zero-one pattern.

When these bits are set to 10b, the preface area is set to the one-zero-one-zero pattern.

When these bits are set to 11b, the preface area is set to all 1s.

#### **RPLEN[3:0] bits (Receive Preface Length)**

The RPLEN[3:0] bits set the preface bit length of the received frames in Manchester mode.

The settable range is 0x0 to 0xF (0d to 15d). The setting of 0x0 disables the receive preface, which is not added. When 0x1 to 0xF is set, the set value is handled as the receive preface bit length.

**RPPAT[1:0] bits (Receive Preface Pattern)**

The RPPAT[1:0] bits set one of the four preface patterns in Manchester mode. For the transmit and receive data when the RPPAT[1:0] bits are set, see [Figure 30.53](#).

When these bits are set to 00b, the preface area is handled as all 0s.

When these bits are set to 01b, the preface area is handled as the zero-one-zero-one pattern.

When these bits are set to 10b, the preface area is handled as the one-zero-one-zero pattern.

When these bits are set to 11b, the preface area is handled as all 1s.

**PFEREN bit (Preface Error Enable)**

The PFEREN bit specifies whether to handle a preface error as an interrupt source.

When it is set to 0, a preface error is not handled as an interrupt source. When it is set to 1, a preface error is handled as an interrupt source.

**SYEREN bit (Receive SYNC Error Enable)**

The SYEREN bit specifies whether to handle a receive SYNC error as an interrupt source.

When it is set to 0, a receive SYNC error is not handled as an interrupt source. When it is set to 1, a receive SYNC error is handled as an interrupt source.

**SBEREN bit (Start Bit Error Enable)**

The SBEREN bit specifies whether to handle a start bit error as an interrupt source.

When it is set to 0, a start bit error is not handled as an interrupt source. When it is set to 1, a start bit error is handled as an interrupt source.

**30.2.13 DCR : Driver Control Register**

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	DENG[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DEAST[4:0]				—	—	—	—	—	—	—	—	DEPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DEPOL	Driver Effective Polarity Select Valid only in Asynchronous mode. 0: The DEN signal is active-high 1: The DEN signal is active-low	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DEAST[4:0]	Driver Assertion Time Valid only in Asynchronous mode. Set the driver assertion time. When DEN = 1, the driver assertion time is inserted in addition to the normal transmission waiting time. Setting DEAST[4:0] = 0x00 is prohibited.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20:16	DENGT[4:0]	Driver Negate Time Valid only in Asynchronous mode. Set the driver negation time. When DEN = 1, the driver negate time is inserted after STOP bit transmission end. Setting DENGT[4:0] = 0x00 is prohibited.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

### DEPOL bit (Driver Effective Polarity Select)

Select the active level of the DEn signal.

### DEAST[4:0] bits (Driver Assertion Time)

Set the driver assertion time (= time from the activation of the DEn (Driver Enable) signal to the start of the start bit). It is expressed in base clock period.

Driver assertion time

= DEAST [4:0] set value × base clock period + transmission waiting time

### DENGT[4:0] bits (Driver Negate Time)

Set the driver negation time (= time from the end of the last stop bit of the transmitted message until the DEn (Driver Enable) signal is disabled). It is expressed in base clock period.

Driver negate time

= DENGT[4:0] set value × base clock period

If the transmission data is written during the driver negate time, transmit starting operation is different depends on the writing timing. The DEn signal remains valid, and transmission of the start bit may start after the transmission wait time has elapsed. Also, the DEn signal may become invalid once, and start bit transmission may start after the Driver assertion time has elapsed.

## 30.2.14 XCR0 : Simple LIN Control Register 0

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0, 1)  
SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	BCCS[1:0]	—	AEDIE	COFIE	BFDIE	—	—	BCDIE	BFOIE	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PIBS[2:0]			PIBE	CF1DS[1:0]	CFOR E	BFE	—	—	—	—	—	—	—	TCSS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TCSS[1:0]	Timer Count Clock Source Selection Valid in Simple LIN mode. Select the clock source of the timer in Simple LIN module. 0 1: TCLK/4 1 0: TCLK/16 1 1: TCLK/64	R/W <sup>1</sup>
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	BFE	Break Field Enable Set the presence or absence of Break Field of Start Frame. 0: No Break Field 1: With Break Field	R/W <sup>3</sup>

Bit	Symbol	Function	R/W
9	CF0RE	Control Field 0 Enable Set the presence or absence of Control Field 0 of Start Frame 0: No Control Field 0 1: With Control Field 0	R/W <sup>3</sup>
11:10	CF1DS[1:0]	Control Field1 Compare Data Select Select the compare data for Control Field 1. 0 0: Select XCR1.PCF1D[7:0] as the compare data 0 1: Select XCR1.SCF1D[7:0] as the compare data 1 0: Select both XCR1.PCF1D[7:0] and XCR1.SCF1D[7:0] as the compare data 1 1: Setting prohibited	R/W <sup>3</sup>
12	PIBE	Priority Interrupt Bit Enable 0: Priority interrupt bit disabled 1: Priority interrupt bit enabled	R/W <sup>3</sup>
15:13	PIBS[2:0]	Priority Interrupt Bit Select Specify one of bits 0 to 7 of Control Field 1 as the priority interrupt bit. 0 0 0: Bit 0 of Control Field 1 0 0 1: Bit 1 of Control Field 1 0 1 0: Bit 2 of Control Field 1 0 1 1: Bit 3 of Control Field 1 1 0 0: Bit 4 of Control Field 1 1 0 1: Bit 5 of Control Field 1 1 1 0: Bit 6 of Control Field 1 1 1 1: Bit 7 of Control Field 1	R/W <sup>3</sup>
16	BFOIE	Break Field Output Completion Interrupt Enable Select whether to include Break Field output completion as a SCIn_TXI interrupt factor. 0: Break Field output completion is not included in SCIn_TXI interrupt factor 1: Break Field output completion is included in SCIn_TXI interrupt factor	R/W
17	BCDIE	Bus Conflict Detection Interrupt Enable Select whether to output an SCIn_ERI interrupt when a bus collision is detected. 0: Bus conflict detection is not included in SCIn_ERI interrupt factor 1: Bus conflict detection is included in SCIn_ERI interrupt factor	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	BFDIE	Break Field Detection Interrupt Enable Select whether to output a SCIn_BFD interrupt when a Break Field is detected. 0: Break Field detection interrupt disabled 1: Break Field detection interrupt enabled	R/W
21	COFIE	Counter Overflow Interrupt Enable Select whether to include counter overflow as an SCIn_ERI interrupt factor. 0: Counter overflow is not included in SCIn_ERI interrupt factor 1: Counter overflow is included in SCIn_ERI interrupt factor	R/W
22	AEDIE	Active Edge Detection Interrupt Enable Select whether to output an SCIn_AED interrupt when a valid edge is detected. 0: Active edge detection interrupt disabled 1: Active edge detection interrupt enabled	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
25:24	BCCS[1:0]	Bus Conflict Detection Clock Selection Select the sampling clock for the bus conflict detection circuit. When CCR2.ABCS = 1, setting BCCS[1] = 1 is prohibited. 0 0: Base clock <sup>*2</sup> 0 1: Base clock/2 1 0: Base clock/4 1 1: Setting prohibited	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. You can rewrite TCSS[1:0] only when the timer is stopped (TCST = 0 and SDST = 0 and BMEN = 0)

Note 2. Base clock: 1/16 period of 1 bit period when CCR2.ABCS = 0, 1/8 period of 1 bit period when CCR2.ABCS = 1.

Note 3. This bit is a setting bit required for the start frame reception operation. Rewrite this bit when the start frame reception or transmission is not in progress (XCR1.SDST = 0 and XCR1.TCST = 0).

**TCSS[1:0] bits (Timer Count Clock Source Selection)**

Select clock source of timer in Simple LIN module.

**BFE bit (Break Field Enable)**

Set the presence or absence of Break Field of Start Frame.

**CF0RE bit (Control Field 0 Enable)**

Set the presence or absence of Control Field 0 of Start Frame.

**CF1DS[1:0] bits (Control Field1 Compare Data Select)**

Select the compare data for Control Field 1.

**PIBE bit (Priority Interrupt Bit Enable)**

Select whether to enable priority interrupt bit comparison of Control Field 1. When the PIBE bit is 1, regardless of the XCR1.CF1CE [7:0] setting value, the bit specified in PIBS [2:0] is compared with the primary comparison data for Control Field 1 (XCR1.PCF1D [7:0]).

**PIBS[2:0] bits (Priority Interrupt Bit Select)**

Specify bit N (N = 0-7) of Control Field 1 as the priority interrupt bit.

**BFOIE bit (Break Field Output Completion Interrupt Enable)**

Select whether to include Break Field output completion as a SCIn\_TXI interrupt factor. Set CCR0.TIE = 1 and CCR3.MOD [1:0] = 110b, to output SCIn\_TXI upon completion of Break Field output.

**BCDIE bit (Bus Conflict Detection Interrupt Enable)**

Select whether to output an SCIn\_ERI interrupt when a bus collision is detected. In Simple LIN mode (CCR3.MOD [1:0] = 110b), SCIn\_ERI output control is performed with the BCDIE bit. When CCR3.MOD [1:0] = 110b and BCDIE = 1, an SCIn\_ERI interrupt is issued when a bus collision is detected even if CCR0.RIE = 0.

**COFIE bit (Counter Overflow Interrupt Enable)**

Select whether to include counter overflow as an SCIn\_ERI interrupt factor. Set CCR0.RIE = 1 and CCR3.MOD [1:0] = 110b are required to output SCIn\_ERI upon counter overflow.

**AEDIE bit (Active Edge Detection Interrupt Enable)**

Select whether to output an SCIn\_AED interrupt when a valid edge is detected. To output SCIn\_AED with valid edge detection, XCR1.BMEN = 1 and CCR3.MOD [1:0] = 110b must be set.

**BCCS[1:0] bits (Bus Conflict Detection Clock Selection)**

Select the sampling clock for the bus conflict detection circuit.

**30.2.15 XCR1 : Simple LIN Control Register 1**

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0, 1)  
SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CF1CE[7:0]								SCF1D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PCF1D[7:0]							—	—	BMEN	SDST	—	—	—	TCST	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Bit	Symbol	Function	R/W
0	TCST	Break Field Output Timer Count Start Trigger 0: Break Field output timer count stopped 1: Break Field output timer count start	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	SDST	Start Frame Detection Enable 0: Start Frame/Break Field detection disabled 1: Start Frame/Break Field detection enabled Do not set this bit and TCST bit to 1 at the same time.	R/W
5	BMEN	Bit Rate Measurement Enable Set this bit to 1 simultaneously with the SDST bit. This bit can be set to 0 at any timing. 0: Bit rate measurement disabled 1: Bit rate measurement enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PCF1D[7:0]	Priority Compare Data for Control Field 1 The priority compare data for Control Field 1.	R/W
23:16	SCF1D[7:0]	Secondary Compare Data for Control Field 1 The secondary compare data for Control Field 1.	R/W
31:24	CF1CE[7:0]	Control Field 1 Compare Bit Enable Select whether to compare bit N of Control Field 1. (N = 0 -7) 0: Control Field 1 bit N compare disabled 1: Control Field 1 bit N compare enabled	R/W

Note: S-TYPE-3, P-TYPE-3

### TCST bit (Break Field Output Timer Count Start Trigger)

[Clearing condition]

- When 0 is written to TCST. Break Field output timer count is stopped and TXDn output becomes idle level.
- When Break Field output for the period set in XCR2.BFLW [15: 0] is completed.

[Setting condition]

- When 1 is written to TCST. Start Break Field output from TXDn. Holds 1 during Break Field output.

### SDST bit (Start Frame Detection Enable)

When 1 is written to the SDST bit, Start Frame detection starts. When XCR0.BFE = 1 is set, Break Field can be detected during Start Frame is detected and after Start Frame is detected. When XCR0.BFE = 0 is set, Break Field is not detected.

When 0 is written to the SDST bit, Start Frame detection and Break Field detection are stopped. However, if XSR0.RXDSF = 0 at the time of stop, it is not possible to stop data reception of the SCI core with this bit. Write 0 to CCR0.RE to stop the reception operation or perform reception completion processing (CSR.RDRF clear or RDR read) after reception is completed.

### BMEN bit (Bit Rate Measurement Enable)

Set this bit to 1 simultaneously with the SDST bit. When the BMEN bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured.

### PCF1D[7:0] bits (Priority Compare Data for Control Field 1)

Set the priority compare data for Control Field 1.

### SCF1D[7:0] bits (Secondary Compare Data for Control Field 1)

Set the secondary compare data for Control Field 1.

### CF1CE[7:0] bits (Control Field 1 Compare Bit Enable)

Select whether to compare bit N of Control Field 1. (N = 0 - 7)

When all of the CF1CE[7:0] bits are set to 0 (CF1CE[7:0] = 0x00), it is always determined that Control Field 1 matches when reception is completed, and XSR0.CF1MF is set. This bit is a comparison enable with PCF1D[7:0] or SCF1D[7:0], and is not a priority interrupt bit comparison enable.

### 30.2.16 XCR2 : Simple LIN Control Register 2

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0, 1)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BFLW[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CF0CE[7:0]								CF0D[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CF0D[7:0]	Control Field 0 Compare Data The compare data for Control Field 0.	R/W
15:8	CF0CE[7:0]	Control Field 0 Compare Bit Enable Select whether to compare bit N of Control Field 0. (N = 0 - 7) 0: Control Field 0 bit N compare disabled 1: Control Field 0 bit N compare enabled	R/W
31:16	BFLW[15:0]	Break Field Length Setting This register sets the Break Field length. The Break Field length is (BFLW [15:0] setting value + 1) × clock of the timer. The upper limit for setting this register is 0xFFFE. Setting is prohibited for 0xFFFF.	R/W

Note: S-TYPE-3, P-TYPE-3

#### CF0D[7:0] bits (Control Field 0 Compare Data)

The compare data for Control Field 0.

#### CF0CE[7:0] bits (Control Field 0 Compare Bit Enable)

Select whether to compare bit N of Control Field 0. (N = 0 - 7)

When all of the CF0CE[7:0] bits are set to 0 (CF0CE[7:0] = 0x00), it is always determined that Control Field 0 matches when reception is completed, and XSR0.CF0MF is set.

#### BFLW[15:0] bits (Break Field Length Setting)

BFLW[15:0] are 16-bit Break Field length setting bits and the initial value is 0xFFFE.

Set the Break Field length to 1 frame or more. The LIN standard stipulates that the Break Field length is 13 bits or more.

When the Break Field is sending, writing 1 to TCST causes SCI to output the Break Field on TXDn. Up-counting is performed with the clock of the timer selected by XCR0.TCSS[1:0]. When the count value matches the value set in this register, up-counting is stopped and Break Field output from TXDn is also stopped.

When the Break Field is receiving, writing 1 to SDST enables Start Frame detection. SCI starts counting from the negative edge of RXDn. The clock of the timer is selected by XCR0.TCSS[1:0].

When the count value matches the value set in the XCR2, a Break Field has been detected. Up-counting continues until the next valid edge or counter overflow.

## 30.2.17 CSR : Common Status Register

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x48

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDRF	TEND	TDRE	FER	PER	MFF	—	ORER	—	—	—	—	—	DFER	DPER	DCMF
Value after reset:	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXDMON	—	—	—	—	—	—	—	—	—	—	ERS	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R
4	ERS	Error Signal Status Flag Valid only in Smart card interface mode. 0: Error signal low not responded 1: Error signal low responded	R
14:5	—	These bits are read as 0.	R
15	RXDMON	Serial Input Data Monitor Bit The state of the RXDn pin without synchronizing by bus clock is shown. 0: When RINV is 0, RXDn pin is the low level. When RINV is 1, RXDn pin is the high level. 1: When RINV is 0, RXDn pin is the high level. When RINV is 1, RXDn pin is the low level.	R
16	DCMF	Data Compare Match Flag Valid only in Asynchronous mode. 0: No matched 1: Matched	R
17	DPER	Data Compare Match Parity Error Flag Valid only in Asynchronous mode. 0: No parity error occurred at address match detection 1: A parity error has occurred at address match detection	R
18	DFER	Data Compare Match Framing Error Flag Valid only in Asynchronous mode. 0: No framing error occurred at address match detection 1: A framing error has occurred at address match detection	R
23:19	—	These bits are read as 0.	R
24	ORER	Overrun Error Flag 0: No overrun error occurred 1: An overrun error has occurred	R
25	—	This bit is read as 0.	R
26	MFF	Mode Fault Error Flag Valid only in Simple SPI mode. 0: No mode fault error 1: Mode fault error	R
27	PER	Parity Error Flag 0: Non-FIFO selected (CCR3.FM = 0): No parity error occurred FIFO selected (CCR3.FM = 1): No parity error in all received data in receive-FIFO 1: Non-FIFO selected (CCR3.FM = 0): A parity error has occurred FIFO selected (CCR3.FM = 1): One or more parity errors occurred in received data in receive-FIFO	R

Bit	Symbol	Function	R/W
28	FER	Framing Error Flag 0: Non-FIFO selected (CCR3.FM = 0): No framing error occurred FIFO selected (CCR3.FM = 1): No framing error in all received data in receive-FIFO 1: Non-FIFO selected (CCR3.FM = 0): A framing error has occurred FIFO selected (CCR3.FM = 1): One or more framing errors occurred in received data in receive-FIFO	R
29	TDRE	Transmit Data Empty Flag 0: Non-FIFO selected (CCR3.FM = 0): Transmit data is in TDR register FIFO selected (CCR3.FM = 1): The quantity of transmit data written in transmit-FIFO exceeds the specified transmit triggering number. 1: Non-FIFO selected (CCR3.FM = 0): No transmit data is in TDR register FIFO selected (CCR3.FM = 1): The quantity of transmit data written in transmit-FIFO is equal to or less than the specified transmit triggering number.	R
30	TEND	Transmit End Flag 0: A character is being transmitted or standing by for transmission. 1: Character transfer has been completed, or sending Break Field.	R
31	RDRF	Receive Data Full Flag 0: Non-FIFO selected (CCR3.FM = 0): No received data is in RDR register FIFO selected (CCR3.FM = 1): The quantity of receive data written in receive-FIFO falls below the specified receive triggering number. 1: Non-FIFO selected (CCR3.FM = 0): Received data is in RDR register FIFO selected (CCR3.FM = 1): The quantity of receive data written in receive-FIFO is equal to or greater than the specified receive triggering number.	R

Note: S-TYPE-3, P-TYPE-3

### ERS bit (Error Signal Status Flag)

[Setting condition]

- When an error signal low is sampled.

[Clearing condition]

- When write 1 to CFCLR.ERSC.

### DCMF bit (Data Compare Match Flag)

The DCMF bit indicates that SCI detects the match to the comparison data (CCR4.CMPD) with receive data.

Clearing the CCR0.RE bit to 0 does not affect the DCMF flag, which retains its previous state.

[Setting condition]

- Matched to the comparison data (CCR4.CMPD) with receive data, while CCR0.DCME = 1.

[Clearing condition]

- When write 1 to CFCLR.DCMFC.

### DPER bit (Data Compare Match Parity Error Flag)

The DPER bit indicates that a parity error occurred at address match detection (reception data match detection).

Clearing the CCR0.RE bit to 0 does not affect the DPER flag, which retains its previous state.

[Setting condition]

- When a parity error is detected by the frame in which an address match was detected.

[Clearing condition]

- When writing 1 to CFCLR.DPERC.

### DFER bit (Data Compare Match Framing Error Flag)

The DFER bit indicates that a framing error occurred at address match detection (reception data match detection).

Clearing the CCR0.RE bit to 0 does not affect the DFER flag, which retains its previous state.

[Setting condition]

- When a stop bit of the frame in which an address match was detected is 0.  
When it is a 2-stop mode, the 1st bit of the stop bit only determines whether it is 1 and does not check the 2nd bit of the stop bit.

[Clearing condition]

- When writing 1 to CFCLR.DFERC.

### ORER bit (Overrun Error Flag)

The ORER bit indicates that an overrun error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the ORER flag, which retains its previous state. In Simple IIC mode, the ORER bit is not used.

[Setting condition with non-FIFO mode (CCR3.FM = 0)]

- When the next data is received before reading out RDR with no-error reception data stored in RDR.  
In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, reception data is not forwarded to RDR register.

Note: In Clock synchronous mode and Simple SPI mode, serial reception is stopped.

[Setting condition with FIFO mode (CCR3.FM = 1)]

- When the next serial reception is completed while the receive FIFO is full of 16 receive data.

[Clearing condition]

- When write 1 to CFCLR.ORERC.

### MFF bit (Mode Fault Error Flag)

The MFF bit indicates mode fault errors. In a multi-master configuration, determine the mode fault error occurrence by reading the MFF flag.

[Setting condition]

- Input on the SSn pin being at the low level during master operation (CCR3.CKE[1:0] = 00 or 01) in Simple SPI mode.

[Clearing condition]

- When writing 1 to CFCLR.MFFC.

### PER bit (Parity Error Flag)

The PER bit indicates that a parity error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the PER flag, which retains its previous state.

In Clock synchronous mode, Simple SPI mode, and Simple IIC mode, the PER bit is not used.

[Setting condition]

- When a parity error is detected during reception. In FIFO select mode, when one or more parity error is detected in receive-FIFO data.  
In non-FIFO mode, although receive data when the parity error occurs is transferred to RDR, no SCIn\_RXI interrupt request occurs. Note that when the PER flag is set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When writing 1 to CFCLR.PERC.

### FER bit (Framing Error Flag)

The FER bit indicates that a framing error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the FER flag, which retains its previous state.

In Clock synchronous mode, Simple SPI mode, and Simple IIC mode, the FER bit is not used.

## [Setting condition]

- When 0 is sampled as the stop bit during reception. In FIFO select mode, when one or more framing error is detected in receive-FIFO data. In Manchester mode, when both sampling results (1/4 and 3/4 sampling points) are not 1 for 1 stop bit. In Simple LIN mode, even if a condition that changes to 1 occurs when XCR1.SDST = 1, the FER set timing is delayed up to the Break Field evaluation timing at the longest, since it may be a Break Field. If an edge is detected on the RXDn signal before the Break Field evaluation timing, FER is detected. If no edge is detected in the RXDn signal before the Break Field evaluation timing, Break Field is detected.  
In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. In non-FIFO mode, although receive data when the framing error occurs is transferred to RDR, no SCIn\_RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

## [Clearing condition]

- When writing 1 to CFCLR.FERC.

**TDRE bit (Transmit Data Empty Flag)**

## [Non-FIFO selected (CCR3.FM = 0)]

The TDRE bit indicates the presence of transmit data in the TDR register.

The condition of CCR0.TE = 0 has priority over the condition of 0.

If other conditions that become 1 and conditions that become 0 are satisfied at the same time, the TDRE flag is set to 0.

## [Setting condition]

- When CCR0.TE is 0.
- When data is transmitted from the TDR register to the TSR register.

## [Clearing condition]

- When writing 1 to CFCLR.TDREC.
- When the transmission data is written to the TDR register during CCR0.TE = 1.

## [FIFO selected (CCR3.FM = 1)]

The TDRE bit indicates that data has been transferred from the transmit-FIFO (TDR) into the transmit shift register (TSR), the quantity of data in transmit-FIFO has fallen below the specified transmit triggering number.

When the condition which becomes 1 and the condition which becomes 0 are formed at the same time, TDRE flag is 0. After that, when the number of data stored in transmit-FIFO is evaluated, and if that is equal to or less than TTRG value, TDRE is set to 1 after 1 PCLK.

## [Setting condition]

- When the quantity of transmit data written in transmit-FIFO is equal to or less than the specified transmit triggering number\*1.

Note 1. The transmit-FIFO is FIFO register of 16 stages, the maximum number of data that can be written when the TDRE flag is 1 is indicated in 0x10 - FTSR.T[5:0]. Even if more data is written, data is discarded.

## [Clearing condition]

- When writing 1 to CFCLR.TDREC.
- When the transmission data is written to transmit-FIFO by the DTC or DMAC (the last block transfer of block transfer).

**TEND bit (Transmit End Flag)**

## [Non-FIFO selected (CCR3.FM = 0), and not Smart card interface mode (CCR3.MOD[2:0] ≠ 001)]

The TEND bit indicates completion of transmission.

## [Setting condition]

- When CCR0.TE is 0.
- When the CCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted.

- When the TDR register is not updated at the end of DE negate time with DE control function enable (CCR3.DEN = 1).
- When Break Field is sending.

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.
- When writing 1 to CFCLR.TDREC during CCR0.TE = 1.

[Non-FIFO selected (CCR3.FM = 0), and Smart card interface mode (CCR3.MOD [2:0] = 001)]

With no error signal from the receiving side, this bit is set to 1 when additional data for transfer is ready to be transferred to the TDR register.

[Setting condition]

- When CCR0.TE is 0.
- When the CCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated. The set timing is determined by register settings as listed below.
  - When GM = 0 and BLK = 0, 12.5etu after the start of transmission
  - When GM = 0 and BLK = 1, 11.5etu after the start of transmission
  - When GM = 1 and BLK = 0, 11.0etu after the start of transmission
  - When GM = 1 and BLK = 1, 11.0etu after the start of transmission

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.
- When writing 1 to CFCLR.TDREC during CCR0.TE = 1.

[FIFO selected (CCR3.FM = 1)]

The TEND bit indicates that the transmit-FIFO does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

[Setting condition]

- TEND is set to 1 when transmit-FIFO does not contain transmit data when the last bit of a one-byte serial character is transmitted.
- When the TDR register is not updated at the end of DE negate time with DE control function enable (CCR3.DEN = 1).

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.

### RDRF bit (Receive Data Full Flag)

[Non-FIFO selected (CCR3.FM = 0)]

The RDRF bit indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing condition]

- When writing 1 to CFCLR.RDRFC.
- When the read data is read from the RDR register.

[FIFO selected (CCR3.FM = 1)]

The RDRF bit indicates that receive data has been transferred to the receive FIFO data register (RDR), and the quantity of data in receive-FIFO is equal to or greater than the specified receive triggering number. When FCR.RTRG is set to 0, RDRF is set if the quantity of data in receive-FIFO is greater than or equal to 1.

[Setting condition]

- RDRF is set to 1 when the quantity of receive data in receive-FIFO is equal to or greater than the specified receive triggering number\*1.

Note 1. Since the receive-FIFO is 16 stage FIFO register, the maximum quantity of data that can be read when RDRF is 1 is equivalent to the specified receive data count number (FDR.R[5:0]). If an attempt is made to read after all the data in receive-FIFO has been read, the data is undefined.

[Clearing condition]

- When writing 1 to CFCLR.RDRFC.
- When the reception data is read from receive-FIFO by the DTC or DMAC (the last block transfer of a block transfer).

When the condition which becomes 1 and the condition which becomes 0 are formed at the same time, RDRF flag is 0. After that, when the number of stored data in receive-FIFO is evaluated, and if that is equal to or greater than RTRG value, RDRF is set to 1 after 1 PCLK.

Note: When Non-FIFO is selected, except when interruption communication, RDRF and TDRE should not be cleared by CFCLR register.

### 30.2.18 ISR : Simple IIC Status Register

Base address: SCLn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
 SCLn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x4C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IICSTI F	—	—	IICAC KR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IICACKR	ACK Reception Data Flag 0: ACK received 1: NACK received	R
1	—	This bit is read as 0.	R
2	—	The read value is undefined.	R
3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag 0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R
5:4	—	The read value is undefined.	R
31:6	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

#### IICACKR bit (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.

The IICACKR flag is updated at the rising of SCLn clock for the ACK/NACK receiving bit.



**IICSTIF bit (Issuing of Start, Restart, or Stop Condition Completed Flag)**

After generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.

When the IICSTIF flag is 1 while an interrupt request is enabled by setting the CCR0.TEIE bit, an STIn request is output.

[ Setting condition ]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the clearing condition takes precedence)

[ Clearing condition ]

- Writing 1 to ICFCLR.IICSTIFC bit
- When operation is not in Simple IIC mode
- Writing 0 to CCR0.TE bit

**30.2.19 FRSR : FIFO Receive Status Register**

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	FNUM[5:0]					—	—	PNUM[5:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	R[5:0]					—	—	—	—	—	—	—	—	DR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	0

Bit	Symbol	Function	R/W
0	DR	Receive Data Ready Flag 0: Receiving is in progress, or no received data remains in the receive-FIFO after a successful reception (receive-FIFO is empty) 1: The following receive data is not received for a fixed period after storing data under the threshold in the receive-FIFO	R
1	—	The read value is undefined.	R
7:2	—	These bits are read as 0.	R
13:8	R[5:0]	Receive-FIFO Data Count Valid in Asynchronous mode (including Multiprocessor mode), Clock synchronous mode, and Simple SPI mode, when CCR3.FM is 1. Indicate the quantity of receive data stored in receive-FIFO	R
15:14	—	These bits are read as 0.	R
21:16	PNUM[5:0]	Parity Error Count Valid only in Asynchronous mode Indicates the quantity of data with a parity error for the receive data stored in the Receive FIFO Data register.	R
23:22	—	These bits are read as 0.	R
29:24	FNUM[5:0]	Framing Error Count Valid only in Asynchronous mode. Indicates the quantity of data with a framing error for the receive data stored in the Receive FIFO Data register.	R
31:30	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

**DR bit (Receive Data Ready Flag)**

The DR bit indicates that the quantity of data stored in the Receive FIFO Data register (RDR) falls below the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in Asynchronous mode. This bit is valid only Asynchronous mode (including Multiprocessor mode) and FIFO selected. In other mode, this bit does not set to 1.

[Setting conditions]

- DR is set to 1 when the following conditions are met:
  - After receive FIFO Data register (RDR) receives less data than the specified receive triggering number, no next data has been received yet after the elapse of 15 etu\*<sup>1</sup> from the last stop bit
  - CSR.FER, PER flags are 0.

Note 1. This is equivalent to one and a half (1.5) frames in the 8-bit format with one stop bit (etu: elementary time unit).

[Clearing conditions]

- When all receive data in the receive FIFO Data (RDR register) is read and 1 is written to FFCLR.DRC.
- When CCR3.FM bit is 0.

**R[5:0] bits (Receive-FIFO Data Count)**

The R[5:0] bits indicate the quantity of receive data stored in the Receive FIFO Data register.

0x00 means no receive data. 0x10 means receive-FIFO is full.

**PNUM[5:0] bits (Parity Error Count)**

The value indicates the quantity of data stored in the Receive FIFO Data register with a parity error.

**FNUM[5:0] bits (Framing Error Count)**

The value indicates the quantity of data stored in the Receive FIFO Data registers with a framing error.

**30.2.20 FTSR : FIFO Transmit Status Register**

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x54

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	T[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	T[5:0]	Transmit-FIFO Data Count Valid in Asynchronous mode (including Multiprocessor mode), Clock synchronous mode, and Simple SPI mode, when CCR3.FM is 1. Indicate the quantity of non-transmit data stored in transmit-FIFO	R
31:6	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

**T[5:0] bits (Transmit-FIFO Data Count)**

The T[5:0] bits indicate the quantity of non-transmitted data stored in transmit-FIFO.

0x00 means no non-transmit data. 0x10 means transmit-FIFO is full.

### 30.2.21 MSR : Manchester Status Register

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0)

Offset address: 0x58

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RSYN C	—	MER	—	SBER	SYER	PFER
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PFER	Preface Error Flag This bit is set when a preface error (pattern mismatch) is detected 0: No preface error detected 1: Preface error detected	R
1	SYER	SYNC Error Flag This bit is set when no edge is detected in the adjustable range during receive retiming 0: No receive SYNC error detected 1: Receive SYNC error detected	R
2	SBER	Start Bit Error Flag This bit is set when a pattern mismatch in the start bit area is detected 0: No start bit error detected 1: Start bit error detected	R
3	—	This bit is read as 0.	R
4	MER	Manchester Error Flag Valid for Manchester mode only 0: No Manchester error occurred 1: Manchester error has occurred	R
5	—	This bit is read as 0.	R
6	RSYNC	Receive SYNC Data Bit It is valid when MCR.SBSEL = 1 in Manchester mode, 0 is read otherwise. 0: The received start bit is DATA SYNC 1: The received start bit is COMMAND SYNC	R
31:7	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

#### PFER bit (Preface Error Flag)

The PFER bit indicates that a preface error was detected when receiving frames in Manchester mode.

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the PFER flag is not affected and retains its previous value.

[Setting condition]

- When detecting a preface error when receiving frames in Manchester mode  
The following operations are performed when a preface error occurs.
  - When MCR.PFEREN = 1  
The received data is not transferred to the RDR register and no SCIn\_RXI interrupt request occurs. Instead, an SCIn\_ERI interrupt request occurs. Note that when the PFER flag is set to 1, the subsequently received data is not transferred to the RDR register.
  - When MCR.PFEREN = 0

The received data is transferred to the RDR register and an SCIn\_RXI interrupt request occurs. An SCIn\_ERI interrupt request is not generated. The subsequent receive operations are not affected even with the PFER flag is set to 1.

[Clearing condition]

- Writing 1 to MFCLR.PFERC.

### SYER bit (SYNC Error Flag)

The SYER bit indicates that a receive SYNC error was detected when receiving frames in Manchester mode with MCR.ERTEN = 1 (Manchester edge retiming enabled).

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the SYER flag is not affected and retains its previous value.

[Setting condition]

- When detecting a receive SYNC error when receiving frames in Manchester mode  
The following operations are performed when a receive SYNC error occurs.
  - When MCR.SYEREN = 1  
Although the received data is transferred to the RDR register, no SCIn\_RXI interrupt request occurs. Instead, an SCIn\_ERI interrupt request occurs. Note that when the SYER flag is set to 1, the subsequently received data is not transferred to the RDR register.
  - When MCR.SYEREN = 0  
The received data is transferred to the RDR register and an SCIn\_RXI interrupt request occurs. An SCIn\_ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SYER flag is set to 1.

[Clearing condition]

- Writing 1 to MFCLR.SYERC.

### SBER bit (Start Bit Error Flag)

The SBER bit indicates that a start bit error was detected when receiving frames in Manchester mode.

Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the SBER flag is not affected and retains its previous value.

[Setting condition]

- When detecting a start bit error when receiving frames in Manchester mode  
The following operations are performed when a start bit error occurs.
  - When MCR.SBEREN = 1  
The received data is not transferred to the RDR register and no SCIn\_RXI interrupt request occurs. Instead, an SCIn\_ERI interrupt request occurs. Note that when the SBER flag is set to 1, the subsequently received data is not transferred to the RDR register.
  - When MCR.SBEREN = 0  
The received data is transferred to the RDR register and an SCIn\_RXI interrupt request occurs. An SCIn\_ERI interrupt request is not generated. The subsequent receive operations are not affected even with the SBER flag is set to 1.

[Clearing condition]

- Writing 1 to MFCLR.SBERC.

### MER bit (Manchester Error Flag)

When data is received in Manchester mode, Manchester error is detected and displayed. Even when the RE bit in CCR0 is set to 0 (serial reception is disabled), the MER flag is not affected and retains its previous value.

[Setting conditions]

- When receiving in Manchester mode and detecting Manchester code error in data area of received frame.

Received data when an error occurs is transferred to the RDR register, but the SCIn\_RXI interrupt request is not generated and the SCIn\_ERI interrupt request is generated.

When the Manchester error flag is set to 1, subsequent receive data is not transferred to the RDR register.

For details on Manchester error, [section 30.5.11. Errors in Manchester Mode](#).

[Clearing condition]

- Writing 1 to MFCLR.MERC.

### RSYNC bit (Receive SYNC Data Bit)

When Manchester mode (CCR3.MOD[2:0] = 101b) and MCR.SBSEL = 1, this bit indicates the type of SYNC of the received start bit. For other settings, it is fixed to 0.

### 30.2.22 XSR0 : Simple LIN Status Register 0

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0, 1)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x5C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CF1RD[7:0]								CF0RD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AEDF	COF	PIBDF	CF1MF	CF0MF	BFDF	BCDF	BFOF	—	—	—	—	—	—	RXDSF	SFSF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SFSF	Start Frame Status Flag 0: Start Frame detection disabled or Start Frame detection complete 1: Before Start Frame detection or during detection	R <sup>1</sup>
1	RXDSF	RxDn Input Status Flag 0: RxDn input to SCI is enabled 1: RxDn input to SCI is disabled	R <sup>1</sup>
7:2	—	These bits are read as 0.	R
8	BFOF	Break Field Output Completion Flag 0: Break Field is being output or is not being output 1: Completed BF output	R
9	BCDF	Bus Conflict Detection Flag 0: No bus collision detected 1: Detected a bus collision	R
10	BFDF	Break Field Detection Flag 0: Break Field is not detected 1: Break Field is detected	R
11	CF0MF	Control Field 0 Compare Match Flag 0: Control Field 0 received data does not match the setting data 1: Control Field 0 received data matches the setting data	R
12	CF1MF	Control Field 1 Compare Match Flag 0: Control Field 1 received data does not match the setting data 1: Control Field 1 received data matches the setting data	R
13	PIBDF	Priority Interrupt Bit Detection Flag 0: Priority interrupt bit is not detected 1: Priority interrupt bit is detected	R
14	COF	Counter Overflow Flag 0: Break Field detection counter has not overflowed 1: Break Field detection counter overflowed	R

Bit	Symbol	Function	R/W
15	AEDF	Active Edge Detection Flag 0: No valid edge detected 1: Detected a valid edge	R
23:16	CF0RD[7:0]	Control Field 0 received data Control Field 0 received data.	R
31:24	CF1RD[7:0]	Control Field 1 received data Control Field 1 received data.	R

Note: S-TYPE-3, P-TYPE-3

Note 1. When PCLK is faster than TCLK, the flag set timing is delayed from the receive data full interrupt (SCIn\_RXI) output. To see this flag under these conditions, wait at least 1 TCLK cycle after the receive data full interrupt (SCIn\_RXI) before reading this register.

### SFSF bit (Start Frame Status Flag)

Indicates whether detect Start Frame is being detected.

[Setting condition]

- When 1 is written to XCR1.SDST.
- When a Break Field is detected in the Control Field 0/Control Field 1/Information Field phase and the transition to the Control Field 0 or Control Field 1 reception state occurs.

[Clearing condition]

- When XCR1.SDST is 0.
- When Start Frame detection is completed.

### RXDSF bit (RXDn Input Status Flag)

Indicates the RXDn input status to the SCI core. When the RXDSF bit is 1, RXDn input is received only by the simple LIN module and the Break Field is detected and is not input to the SCI core.

### BFOF bit (Break Field Output Completion Flag)

Indicates the completion of Break Field output.

The BFOF bit can be cleared to 0 by writing 1 to XFCLR.BFOC.

### BCDF bit (Bus Conflict Detection Flag)

Indicates the detection of bus conflict in Simple LIN transmit operation.

The BCDF bit can be cleared to 0 by writing 1 to XFCLR.BCDC.

### BFDF bit (Break Field Detection Flag)

Indicates Break Field detection.

The BFDF bit can be cleared to 0 by writing 1 to XFCLR.BFDC.

### CF0MF bit (Control Field 0 Compare Match Flag)

Indicates compare match of Control Field 0 and compare data.

The CF0MF bit can be cleared to 0 by writing 1 to XFCLR.CF0MC.

### CF1MF bit (Control Field 1 Compare Match Flag)

Indicates compare match detection of Control Field 1 and compare data.

The CF1MF bit can be cleared to 0 by writing 1 to XFCLR.CF1MC.

### PIBDF bit (Priority Interrupt Bit Detection Flag)

Indicates compare match detection of Control Field 1 and priority interrupt bit.

The PIBDF bit can be cleared to 0 by writing 1 to XFCLR.PIBDC.

**COF bit (Counter Overflow Flag)**

Indicates that the 16-bit counter in the simple LIN module has overflowed.

The COF bit can be cleared to 0 by writing 1 to XFCLR.COFC.

**AEDF bit (Active Edge Detection Flag)**

Indicates active edge detection.

The AEDF bit can be cleared to 0 by writing 1 to XFCLR.AEDC and when read out XSR1.TCNT[15:0].

**CF0RD[7:0] bits (Control Field 0 received data)**

Stores the received data with a Control Field 0 match detected.

**CF1RD[7:0] bits (Control Field 1 received data)**

Stores the received data with a Control Field 1 match detected.

**30.2.23 XSR1 : Simple LIN Status Register 1**

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0, 1)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x60

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TCNT[15:0]	Timer Count Capture Value Stores the 16-bit counter capture value. The initial value is 0000.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

**TCNT[15:0] bits (Timer Count Capture Value)**

Stores the capture value of the 16-bit counter of the Simple LIN module.

- When sending Start Frame  
The XSR1 holds the previous value.
- When receiving Start Frame with bit rate measurement disabled  
If a Break Field is detected in the Break Field detection state (see [Figure 30.81](#)), the Break Field length is captured and held (counter value is captured at the rising edge of RXDn). If a Break Field is detected in a state other than the Break Field detection state, the previous value is retained.  
If the counter overflows, it is not captured.
- When receiving Start Frame with bit rate measurement enabled  
The count value is captured and held at the valid edge (both RXDn edges). However, in the Break Field detection state, the count value is not captured even if a valid edge occurs. Counter capture value retention is canceled by the XSR1 read. Even if a valid edge occurs before reading, the counter value is not captured.

## 30.2.24 CFCLR : Common Flag Clear Register

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDRF C	—	TDRE C	FERC	PERC	MFFC	—	ORER C	—	—	—	—	—	DFER C	DPER C	DCMF C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ERSC	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	The write value should be 0.	W
4	ERSC	ERS Clear Bit Setting this bit to 1 clears the CSR.ERS bit. The read value is always 0.	W
15:5	—	The write value should be 0.	W
16	DCMFC	DCMF Clear Bit Setting this bit to 1 clears the CSR.DCMF bit. The read value is always 0.	W
17	DPERC	DPER Clear Bit Setting this bit to 1 clears the CSR.DPER bit. The read value is always 0.	W
18	DFERC	DFER Clear Bit Setting this bit to 1 clears the CSR.DFER bit. The read value is always 0.	W
23:19	—	The write value should be 0.	W
24	ORERC	ORER Clear Bit Setting this bit to 1 clears the CSR.ORER bit. The read value is always 0.	W
25	—	The write value should be 0.	W
26	MFFC	MFF Clear Bit Setting this bit to 1 clears the CSR.MFF bit. The read value is always 0.	W
27	PERC	PER Clear Bit Setting this bit to 1 clears the CSR.PER bit. The read value is always 0.	W
28	FERC	FER Clear Bit Setting this bit to 1 clears the CSR.FER bit. The read value is always	W
29	TDREC	TDRE Clear Bit Setting this bit to 1 clears the CSR.TDRE bit. The read value is always 0.	W
30	—	The write value should be 0.	W
31	RDRFC	RDRF Clear Bit Setting this bit to 1 clears the CSR.RDRF bit. The read value is always 0.	W

Note: S-TYPE-3, P-TYPE-3



### 30.2.25 ICFCLR : Simple IIC Flag Clear Register

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x6C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IICSTI FC	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	The write value should be 0.	W
3	IICSTIFC	IICSTIF Clear Bit Setting this bit to 1 clears the ISR.IICSTIF bit. The read value is always 0.	W
31:4	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

### 30.2.26 FFCLR : FIFO Flag Clear Register

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x70

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

Bit	Symbol	Function	R/W
0	DRC	DR Clear Bit Setting this bit to 1 clears the FRSR.DR bit. The read value is always 0.	W
31:1	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

### 30.2.27 MFCLR : Manchester Flag Clear Register

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0)

Offset address: 0x74

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MERC	—	SBER C	SYER C	PFER C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PFERC	PFER Clear Bit Setting this bit to 1 clears the MSR.PFER bit. The read value is always 0.	W
1	SYERC	SYER Clear Bit Setting this bit to 1 clears the MSR.SYER bit. The read value is always 0.	W
2	SBERC	SBER Clear Bit Setting this bit to 1 clears the MSR.SBER bit. The read value is always 0.	W
3	—	The write value should be 0.	W
4	MERC	MER Clear Bit Setting this bit to 1 clears the MSR.MER bit. The read value is always 0.	W
31:5	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

### 30.2.28 XFCLR : Simple LIN Flag Clear Register

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0, 1)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0, 1)

Offset address: 0x78

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AEDC	COFC	PIBDC	CF1M C	CF0M C	BFDC	BCDC	BFOC	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	The write value should be 0.	W
8	BFOC	BFOF Clear Bit Setting this bit to 1 clears the XSR0.BFOF bit. The read value is always 0.	W
9	BCDC	BCDF Clear Bit Setting this bit to 1 clears the XSR0.BCDF bit. The read value is always 0.	W
10	BFDC	BFDF Clear Bit Setting this bit to 1 clears the XSR0.BDFD bit. The read value is always 0.	W
11	CF0MC	CF0MF Clear Bit Setting this bit to 1 clears the XSR0.CF0MF bit. The read value is always 0.	W

Bit	Symbol	Function	R/W
12	CF1MC	CF1MF Clear Bit Setting this bit to 1 clears the XSR0.CF1MF bit. The read value is always 0.	W
13	PIBDC	PIBDF Clear Bit Setting this bit to 1 clears the XSR0.PIBDF bit. The read value is always 0.	W
14	COFC	COFF Clear Bit Setting this bit to 1 clears the XSR0.COF bit. The read value is always 0.	W
15	AEDC	AEDF Clear Bit Setting this bit to 1 clears the XSR0.AEDF bit and cancels holding the XSR1 register. The read value is always 0.	W
31:16	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

### 30.2.29 CESR : Communication Enable Status Register

Base address: SCIn\_B = 0x4035\_8000 + 0x0100 × n (n = 0 to 4, 9)  
 SCIn\_B\_NS = 0x5035\_8000 + 0x0100 × n (n = 0 to 4, 9)

Offset address: 0x1C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TIST	—	—	—	RIST

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	RIST	RE Internal Status 0: RE signal internal state value 0 1: RE signal internal state value 1	R
3:1	—	These bits are read as 0.	R
4	TIST	TE Internal Status 0: TE signal internal state value 0 1: TE signal internal state value 1	R
7:5	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The operation clocks of the communication module and control register can be used asynchronously. Since some control register values are transmitted internally through the synchronization circuit so that they operate correctly even if they are asynchronous, it takes some time for the state to be reflected internally after rewriting the control register.

Communication enable CCR0.TE and CCR0.RE correspond to this register, and when these control bits change from 1 to 0 to rewrite the control bits for the next communication, the TE and RE bits signal it is necessary to rewrite the next control bit after the internal state becomes 0. If a very slow clock is used for the communication module clock, the TE and RE bit states are not reflected internally. At this time, you can check the internal status using this register.

## 30.3 Operation in Asynchronous Mode

Figure 30.6 shows the general format for asynchronous serial communications. One frame consists of a start bit (low level), transmit or receive data, a parity bit, and stop bits (high level). In asynchronous serial communications, the communications line is usually held in the mark state (high level).

The SCI monitors the communications line. When the SCI detects a start bit, it starts serial communication. The detection condition of the start bit changes according to the CCR3.RXDESEL setting. SCI regards space (Low level) as a start bit when CCR3.RXDESEL is 0. SCI regards a fall edge as a start bit when RXDESEL is 1.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and receiver have a double-buffered structure in addition to FIFO mode, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

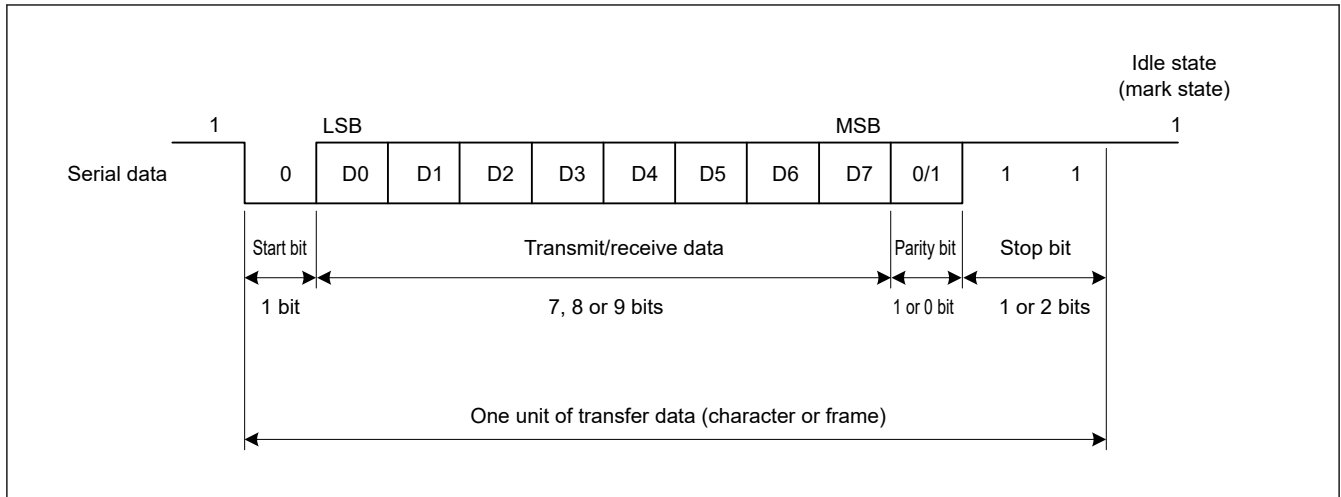


Figure 30.6 Data format in asynchronous serial communications with 8-bit data, parity bit, and 2 stop bits

### 30.3.1 Serial Data Transfer Format

Table 30.28 lists the serial data transfer formats that can be used in Asynchronous mode. Any of 18 transfer formats can be selected with the CCR1 and CCR3 settings. For details on the multi-processor function, see section 30.4. Multi-Processor Communication Function.

Table 30.28 Serial transfer formats in Asynchronous mode (1 of 2)

CCR3		CCR1		CCR3	Serial transfer format and frame length														
CHR[1:0]		PE	MP	STP	1	2	3	4	5	6	7	8	9	10	11	12	13		
0	0	0	0	0	ST	9-bit data									SP				
0	0	0	0	1	ST	9-bit data									SP	SP			
0	0	1	0	0	ST	9-bit data									P	SP			
0	0	1	0	1	ST	9-bit data									P	SP	SP		
1	0	0	0	0	ST	8-bit data								SP					
1	0	0	0	1	ST	8-bit data								SP	SP				
1	0	1	0	0	ST	8-bit data								P	SP				
1	0	1	0	1	ST	8-bit data								P	SP	SP			
1	1	0	0	0	ST	7-bit data							SP						
1	1	0	0	1	ST	7-bit data							SP	SP					
1	1	1	0	0	ST	7-bit data							P	SP					
1	1	1	0	1	ST	7-bit data							P	SP	SP				

**Table 30.28 Serial transfer formats in Asynchronous mode (2 of 2)**

CCR3		CCR1		CCR3	Serial transfer format and frame length												
CHR[1:0]		PE	MP	STP	1	2	3	4	5	6	7	8	9	10	11	12	13
0	0	—	1	0	ST	9-bit data									MPB	SP	
0	0	—	1	1	ST	9-bit data									MPB	SP	SP
1	0	—	1	0	ST	8-bit data								MPB	SP		
1	0	—	1	1	ST	8-bit data								MPB	SP	SP	
1	1	—	1	0	ST	7-bit data							MPB	SP			
1	1	—	1	1	ST	7-bit data							MPB	SP	SP		

ST: Start bit  
 SP: Stop bit  
 P: Parity bit  
 MPB: Multi-processor bit

### 30.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In Asynchronous mode, the SCI operates on a base clock with a frequency of 16 times<sup>\*1</sup> the bit rate.

In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization.<sup>\*2</sup>

Because receive data is sampled on the rising edge of the 8th pulse<sup>\*1</sup> of the base clock, data is latched at the middle of each bit (when sampling timing does not adjust (CCR4.ASEN = 0 or CCR4.ASEN = 1 and CCR4.AST[2:0] = 000b) ), as shown in [Figure 30.7](#) The reception margin in Asynchronous mode is determined by the following formula (1):

$$M = \left| \left( 0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N} (1 + F) \right| \times 100 \text{ [%]} \quad \dots \text{ Formula (1)}$$

Note: M: Reception margin

N: Ratio of bit rate to clock

See [Table 30.6](#) for bit rate to basic clock ratios.

D: Duty cycle of clock (D = 0.5 to 1.0)

L: Frame length (L = 9 to 13)

F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined using the following formula:

$$M = \{0.5 - 1/(2 \times 16)\} \times 100 \text{ (%) = 46.875 \%}$$

This represents the computed value. Renesas recommends a margin of 20% to 30% in system design.

Note 1. This is an example when CCR2.ABCS is 0, CCR2.ABCSE is 0 and CCR2.ABCSE2 is 0. When CCR2.ABCS is 1, CCR2.ABCSE is 0 and CCR2.ABCSE2 is 0, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock. When CCR2.ABCSE is 1 and CCR2.ABCSE2 is 0, a frequency of 6 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 3rd pulse of the base clock. When CCR2.ABCSE is 0 and CCR2.ABCSE2 is 1, a frequency of 4 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 2nd pulse of the base clock.

Note 2. The determination condition of the start bit is as follows.

The function of adjust sampling timing is OFF (ASEN = 0):

The determination condition of a start bit is that Low beyond half bit length continues. It is same as the sampling timing. In [Figure 30.7](#), Low period should be kept over 8-cycles to detect a start bit. If Low period does not keep over 8-cycles, the SCI judges this as a noise. So, the SCI does not start reception and wait start bit.

The function of adjust sampling timing is ON (ASEN = 1):

The determination condition of a start bit is that Low keeps up until the sampling timing. Adjusting the sampling timing forward (AJD = 1) increases the possibility of erroneously determining a noise as the start bit.

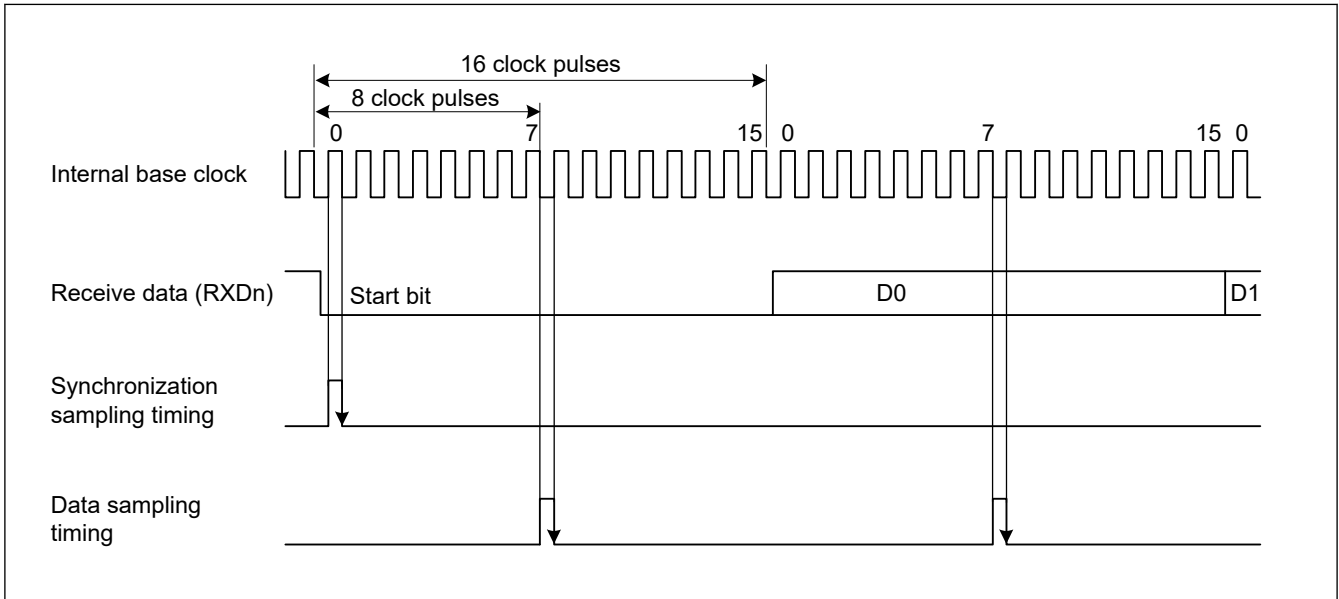


Figure 30.7 Receive data sampling timing in Asynchronous mode

### 30.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the transfer clock of the SCI, based on the CCR3.CKE[1:0] settings.

When an external clock is input to the SCKn pin, the clock frequency must be 16 times the bit rate (when CCR2.ABCS = 0) or 8 times the bit rate (when CCR2.ABCS = 1).

In addition, when an external clock is specified, the base clock of GPTn (n = 3, 4) can be selected by the CCR2.ACS0.

When the SCI uses its internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is configured so that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 30.8.

When the internal clock is selected, the SCKn pin is outputted after setting the CCR0.TE or CCR0.RE bit to 1.

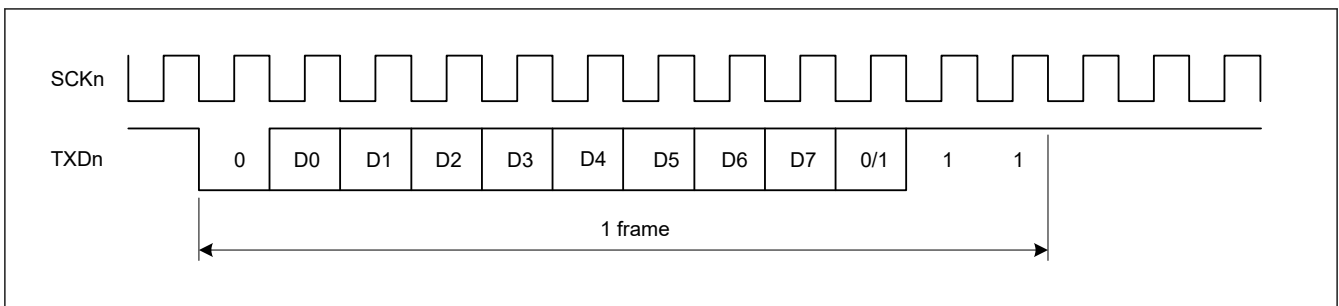


Figure 30.8 Phase relationship between output clock and transmit data in Asynchronous mode when CCR1.PE = 1, CCR3.CHR[1:0] = 10b, MP = 0, and STP = 1

### 30.3.4 Double-Speed Operation and Frequency of 6 or 4 Times the Bit Rate

When the CCR2.ABCS bit is set to 1, the SCI operates on the bit rate twice that of when ABCS is set to 0. When the CCR2.BGDM bit is set to 1, the cycle of the base clock is half and the bit rate is double that of when BGDM is set to 0. When the CCR3.CKE[1] bit is set to 0 and the on-chip baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the SCI to operate at a bit rate four times that when the ABCS and BGDM bits are set to 0.

When CCR2.ABCSE is set to 1 and CCR2.ABCSE2 is set to 0, the number of base clock pulses are 6 during a period of 1 bit, and the base clock frequency is half. And RSCI works 16/3 times of bit rate compared with a case of CCR2.ABCS = 0, CCR2.BGDM = 0, CCR2.ABCSE = 0 and CCR2.ABCSE2 = 0.

When CCR2.ABCSE is set to 1 and CCR2.ABCSE2 is set to 1, the number of base clock pulses are 4 during a period of 1 bit, and the base clock frequency is half. And RSCI works 8 times of bit rate compared with a case of CCR2.ABCS = 0, CCR2.BGDM = 0, CCR2.ABCSE = 0 and CCR2.ABCSE2 = 0.

As shown by Formula (1) in [section 30.3.2. Receive Data Sampling Timing and Reception Margin in Asynchronous Mode](#), the reception margin decreases when CCR2.ABCS is set to 1, CCR2.ABCSE is set to 1 or CCR2.ABCSE2 is set to 1. Therefore, if the desired bit rate can be obtained, it is recommended to use the RSCI with CCR2.ABCS set to 1, CCR2.ABCSE set to 1 and CCR2.ABCSE2 set to 1.

### 30.3.5 CTS and RTS Functions

The CTS function controls transmission using the CTSn pin input. Setting the CCR1.CTSE bit to 1 enables the CTS function. For the functions of CTS and RTS, you can select the alternate setting to set CTSn\_RTSn pin as a multiplexed pin that uses either function with one pin or the dedicated setting that uses each function independently with two pins at CTSn pin for CTSn signal and CTSn\_RTSn pin for RTSn signal. This setting is done with the CCR1.CTSPEN bit.

When the CTS function is enabled, placing a low level on the CTSn\_RTSn pin causes transmission to start.

If FIFO is used and CTSn\_RTSn signal is held high before transmission, transmission will not start, so the number of TDR registers written, and the number of data stored are the same (unlike using clock synchronous FIFO).

Driving the CTSn\_RTSn pin high while transmission is in progress does not affect transmission of the current frame.

In the RTS function, which uses output on the CTSn\_RTSn pin, a low level is output when reception becomes possible. Conditions for output of the low and high levels are shown in this section.

[Conditions for low level output]

Satisfaction of all conditions are listed in this section.

#### Non-FIFO selected

- The value of the CCR0.RE bit is 1
- When the next reception is possible
  - There are no received data yet to be read and not receiving.
  - CSR.ORER, FER, PER flags are all 0

#### FIFO selected

- The value of the CCR0.RE bit is 1
- When the next reception is possible
  - When the quantity of receive data written in receive-FIFO(RDR) are less than the setting value of FCR.RSTRG[4:0]
  - CSR.ORER(RDR.ORER) is 0

[Condition for high level output]

- The conditions for low-level output are not satisfied

### 30.3.6 Address Match (Receive Data Match Detection) Function

The address match function can be used only in Asynchronous mode.

If the CCR0.DCME bit is set to 1<sup>2</sup>, when one frame of data is received, the SCI compares that received data with the data set in CCR4.CMPD. If the SCI detects a match to the comparison data (CCR4.CMPD<sup>1</sup>) with the received data, the SCI can issue the SCIn\_RXI interrupt request.

If the CCR3.MP bit is set to 0, comparison occurs only for valid data in receive format. In multi-processor mode (CCR3.MP bit = 1), if the CCR0.IDSEL bit is set to 1, receive data where the MPB bit is 1 is subject to comparison for address match and receive data where the MPB bit is 0 is always treated as a non-match.

If the CCR0.IDSEL bit is set to 0, SCI performs address match detection regardless of the MPB bit value of the received data.

Until SCI detects a match to the comparison data (CCR4.CMPD\*<sup>1</sup>) with receive data, received data is skipped (discarded), and the SCI cannot detect a parity error or framing error.

When SCI detects a match, the CCR0.DCME bit is automatically cleared, and the CSR.DCMF flag is set to 1. If the CCR0.IDSEL bit is set to 1, the CCR0.MPIE bit is automatically cleared. If CCR0.IDSEL is set to 0, the value of the CCR0.MPIE bit is retained. If the CCR0.RIE bit is set to 1, the SCI issues an SCI<sub>In</sub>\_RXI interrupt request.

If the SCI detects a framing error in the receive data for which a match is detected, the CSR.DFER flag is set to 1, and if the SCI detects a parity error in that frame, the CSR.DPER flag is set to 1. The compared receive data and MPB bit is not stored in the RDR register, and CSR.RDRF remains 0.

After the SCI detects a match, and CCR0.DCME is automatically cleared, the SCI receives the next data continuously based on the current register setting.

When the CSR.DFER or CSR.DPER flag is set, the address match is not performed. Before enabling the address match function, set the CSR.DFER and CSR.DPER flags to 0.

Examples of the address match function are shown in [Figure 30.9](#) and [Figure 30.10](#).

Note 1. This comparative target can select one length of 3 types: CMPD[6:0] with 7-bit length, CMPD[7:0] with 8-bit length, and CMPD[8:0] with 9-bit length.

Note 2. Set the CCR0.DCME bit to 1 before receiving the start bit of the received frame that performs address matching.



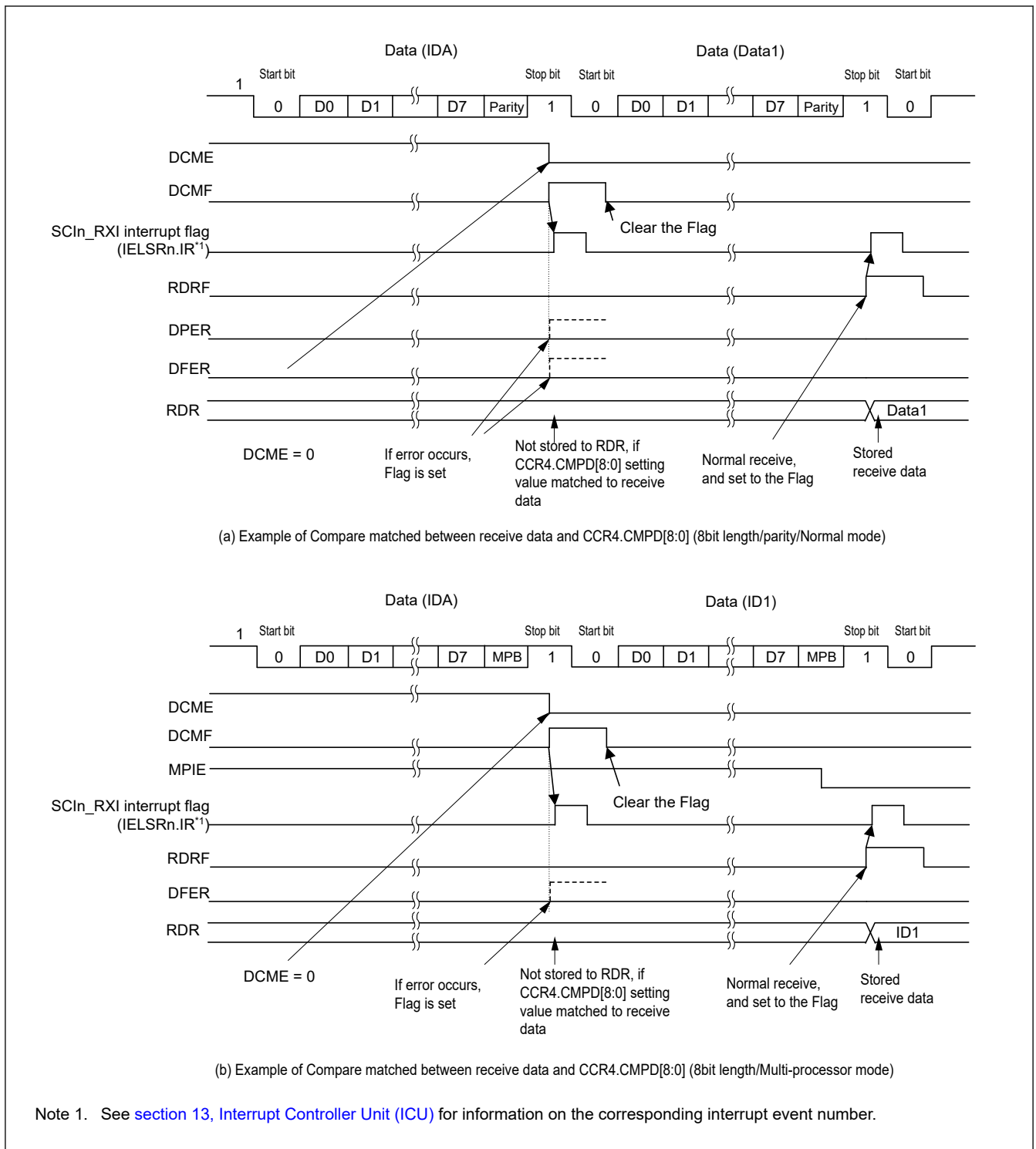


Figure 30.9 Example of address match (1) normal mode and multi-processor mode

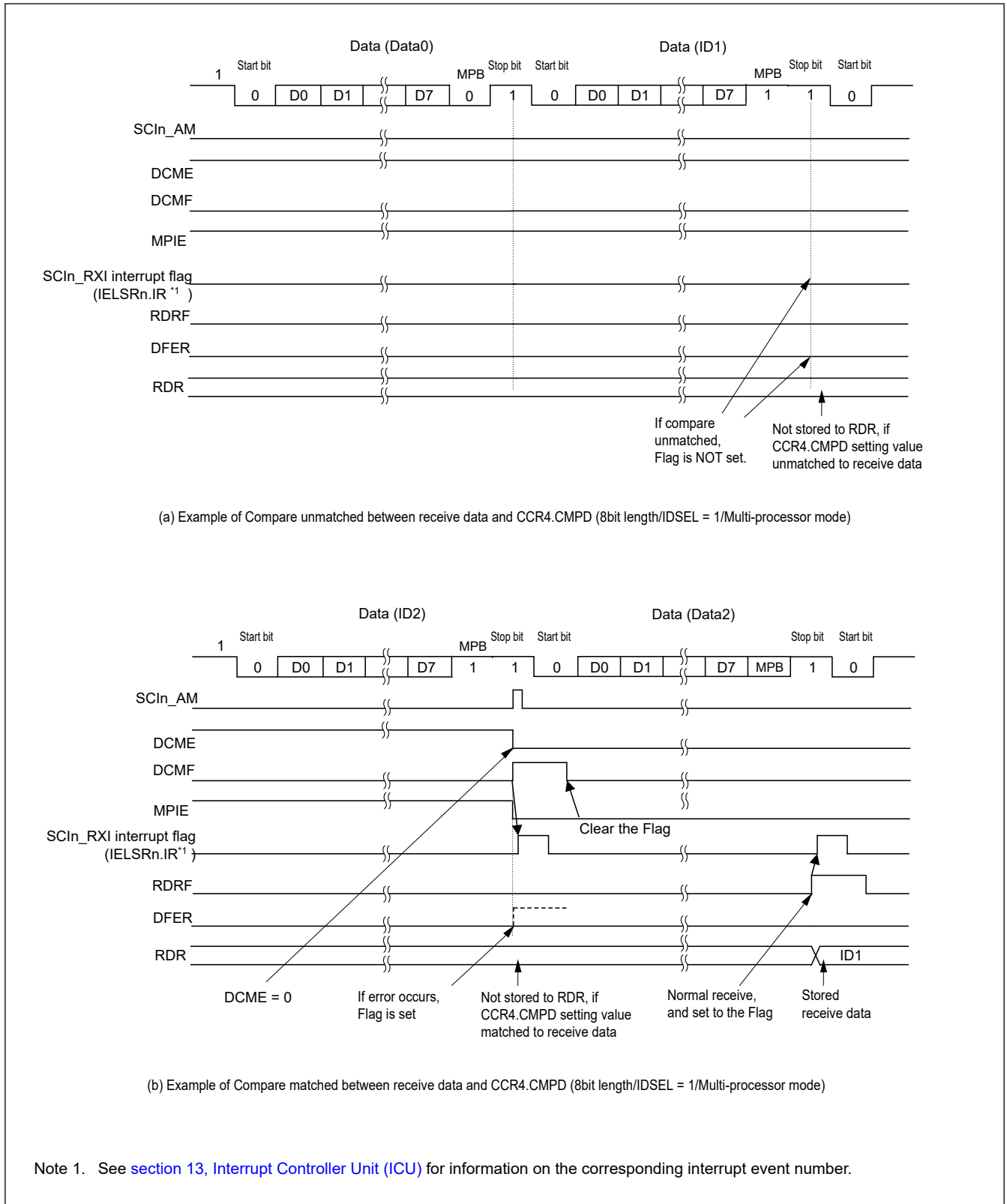


Figure 30.10 Example of address match (2) multi-processor mode

### 30.3.7 SCI Initialization in Asynchronous Mode

Before transmitting and receiving data, start by writing the initial value 0 to the CCR0.TE and CCR0.RE (or writing the initial value to CCR0), then continue through the SCI initialization procedure (select non-FIFO or FIFO) shown in [Table 30.29](#) and [Table 30.30](#). Whenever the operating mode or transfer format is to be changed, the CCR0.TE and CCR0.RE bits must be initialized before the change is made.

When the external clock is used in Asynchronous mode, ensure that the clock signal is supplied during initialization.

Note: Setting the CCR0.RE bit to 0 initializes neither the ORER, FER, RDRF, RDAT, PER, and DR flags in CSR nor RDR. When FIFO selected, even if the TE bit is set as 0, the TEND flag for the selected FIFO buffer is not initialized. Please be also careful at the time of change in the operation mode.

Note: Switching the value of the CCR0.TE bit from 0 to 1 while the CCR0.TIE bit is 1 leads to the generation of an SCIn\_TXI interrupt request.

**Table 30.29 Example flow of SCI initialization in Asynchronous mode with non-FIFO selected**

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set CCR3	Set up following function and communication mode. Driver control function for RS-485, FIFO no-use, Multi-Processor mode, Communication mode (MOD[2:0] = 000b) Transmission / reception format Clock enable (Leave the initial value when outputting the clock) Leave unused bits at their initial values.
4	Set CCR2	Set up the bit-rate-modulation function <sup>*1 *2</sup> , select clock, set bit rate <sup>*2</sup>
5	Set CCR1	Set up the Noise-filter function, the loop-back function, communication pin status, the parity check function, and the CTSn_RTsn function.
6	Set CCR4	Set up the adjust sampling timing function and the adjust transmit timing function.
7	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
8	Set CCR3	Set clock enable bit (CKE[1:0]) at this step when outputting the clock. After this register setting, the clock pin will be in the output state immediately.
9	Set CFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC,FERC,PERC,MFFC,ORERC,DFERC,DPERC,DCMFC,ERSC If it's initialization flow after a reset, you can skip this step.
10	Set CCR0	Set the TE or RE bit to 1. To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Setting the TE and RE bits allows TXDn and RXDn to be used.
11	Initialization completed	—

Note 1. If you do not use the bit rate modulation function, you do not need to set it.

Note 2. If you use an external clock, you do not need to set it.

**Table 30.30 Example flow of SCI initialization in Asynchronous mode with FIFO selected (1 of 2)**

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set CCR3	Set up following function and communication mode. Driver control function for RS-485, FIFO use, Multi-Processor mode, Communication mode (MOD[2:0] = 000b) Transmission / reception format Clock enable (Leave the initial value when outputting the clock) Leave unused bits at their initial values.
4	Set CCR2	Set up the bit-rate-modulation function <sup>*1 *2</sup> , select clock, set bit rate <sup>*2</sup>
5	Set CCR1	Set up the Noise-filter function, the loop-back function, communication pin status, the parity check function, and the CTSn_RTsn function.
6	Set CCR4	Set up the adjust sampling timing function and the adjust transmit timing function.
7	Set FCR	Set the TFRST and RFRST to 1 to empty FIFO. Set the DRES, TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits

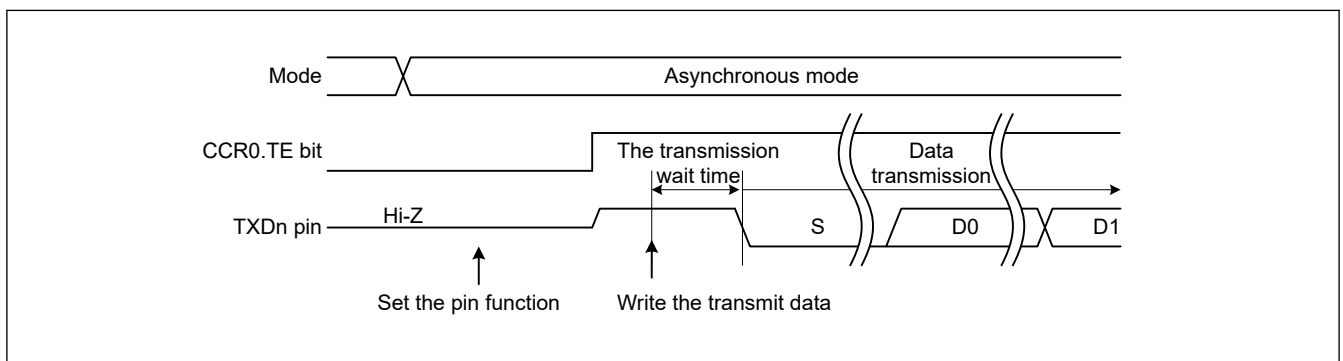
**Table 30.30 Example flow of SCI initialization in Asynchronous mode with FIFO selected (2 of 2)**

No.	Step Name	Description
8	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
9	Set CCR3	If you select a clock output in Asynchronous mode, set the CKE[1:0] bit here. After setting the register, the clock pin will be in the output state immediately. But the clock operates after setting TE or RE to 1.
10	Set CFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC If it's initialization flow after a reset, you can skip this step.
11	Set FFCLR	Write 1 to The FFCLR.BRKC,DRC and clear the corresponding flag. If it's initialization flow after a reset, you can skip this step.
12	Set CCR0	Set the TE or RE bit to 1. To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Setting the TE and RE bits allows TXDn and RXDn to be used.
13	Initialization completed	—

Note 1. If you do not use the bit rate modulation function, you do not need to set it.

Note 2. If you use an external clock, you do not need to set it.

Figure 30.11 shows an example of the timing when data is transmitted after reset is released, and SCI is set to Asynchronous mode according to Table 30.29 or Table 30.30. As shown in the figure, when the pin function is set to the TXDn pin, the CCR0.TE bit is 0, so the pin is high impedance. When transmit data is written after setting the CCR0.TE bit to 1, data transmission starts. There is a transmission wait time from writing TDR to data transmission starts. In Asynchronous mode, TXDn is high during this period.

**Figure 30.11 Data transmission timing example in Asynchronous mode**

### 30.3.8 Serial Data Transmission in Asynchronous Mode

#### (1) Non-FIFO selected

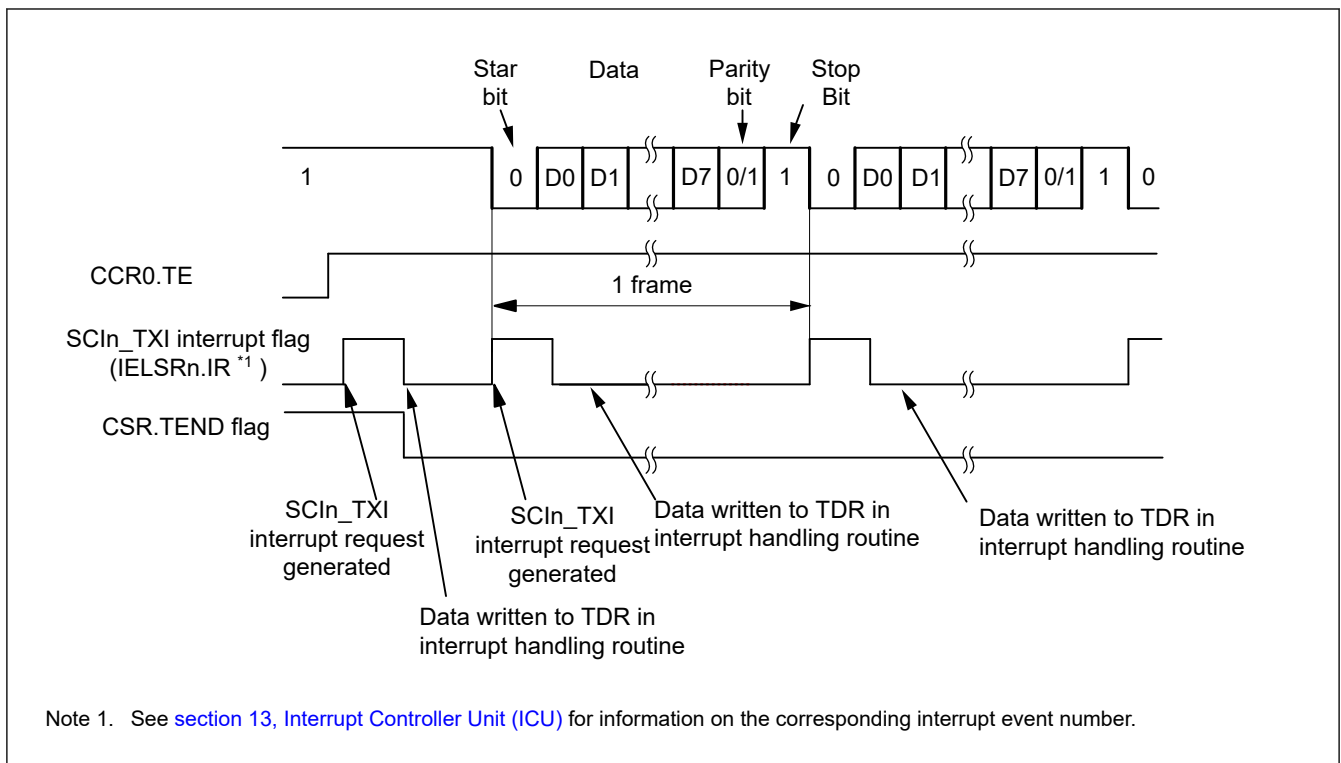
Figure 30.12, Figure 30.13, Figure 30.14 and Figure 30.15 show examples of serial transmission in Asynchronous mode.

In serial transmission, the SCI operates as described in this section.

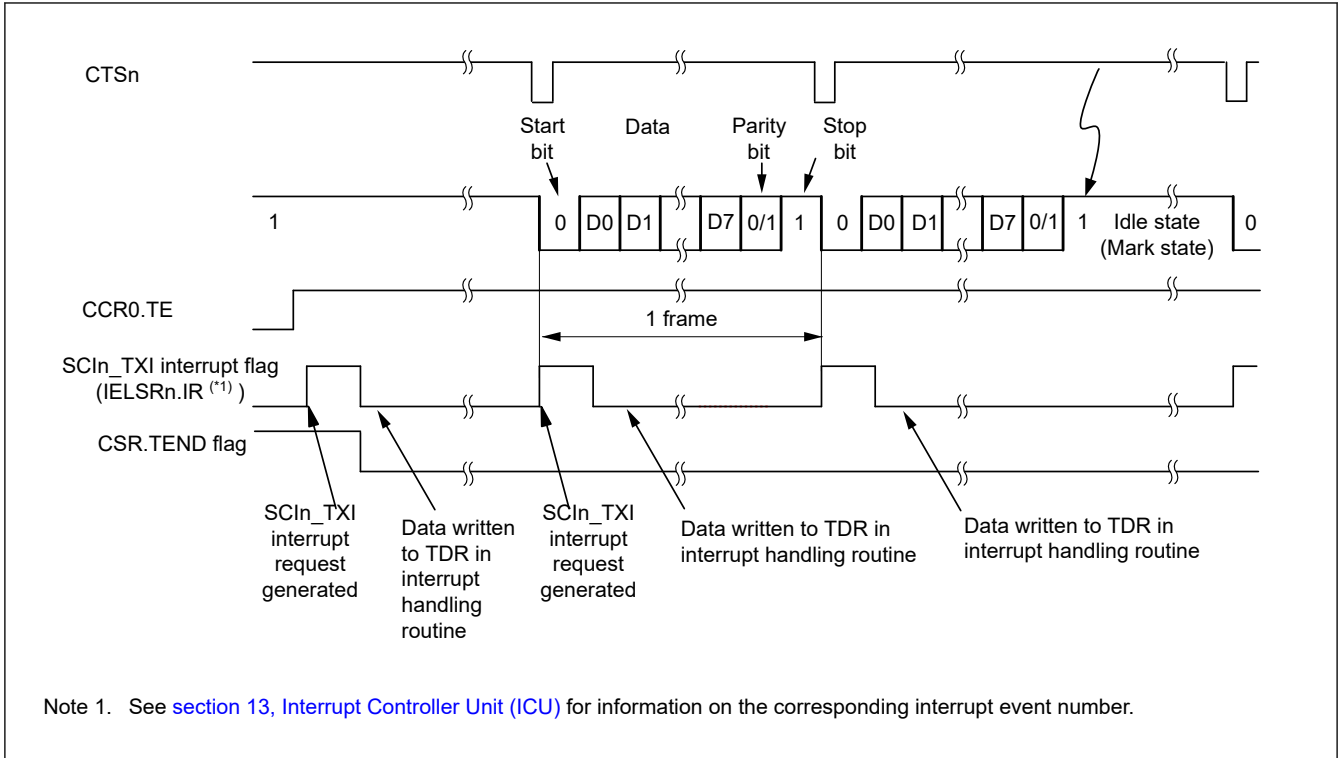
- The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn\_TXI interrupt handling routine.  
The SCIn\_TXI interrupt request at the beginning of transmission is generated when the CCR0.TE and CCR0.TIE bits are set to 1 simultaneously by a single instruction.
- Transmission starts after the CCR1.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn\_RTsn pin causes data transfer from the TDR register to the TSR register. If the CCR0.TIE bit is 1, an SCIn\_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to the TDR register in the SCIn\_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn\_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 (SCIn\_TXI interrupt requests are disabled) and the CCR0.TEIE bit to 1 (an SCIn\_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn\_TXI requests.
- Data is sent from the TXDn pin in the following order:
  - Start bit

- Transmit data
  - Parity bit or multi-processor bit (can be omitted depending on the format)
  - Stop bit
4. The SCI checks for update of the TDR register on output of the stop bit.
  5. When the TDR register is updated, setting the CCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn\_RTSn pin causes transfer of the next transmit data from the TDR register to the TSR register and transmission of the stop bit, after which serial transmission of the next frame starts.
  6. If the TDR register is not updated, the CSR.TEND flag is set to 1, the stop bit is sent, and the mark state is entered, in which 1 is output. If the CCR0.TEIE bit is 1, the CSR.TEND flag is set to 1 and an SCIn\_TEI interrupt request is generated.

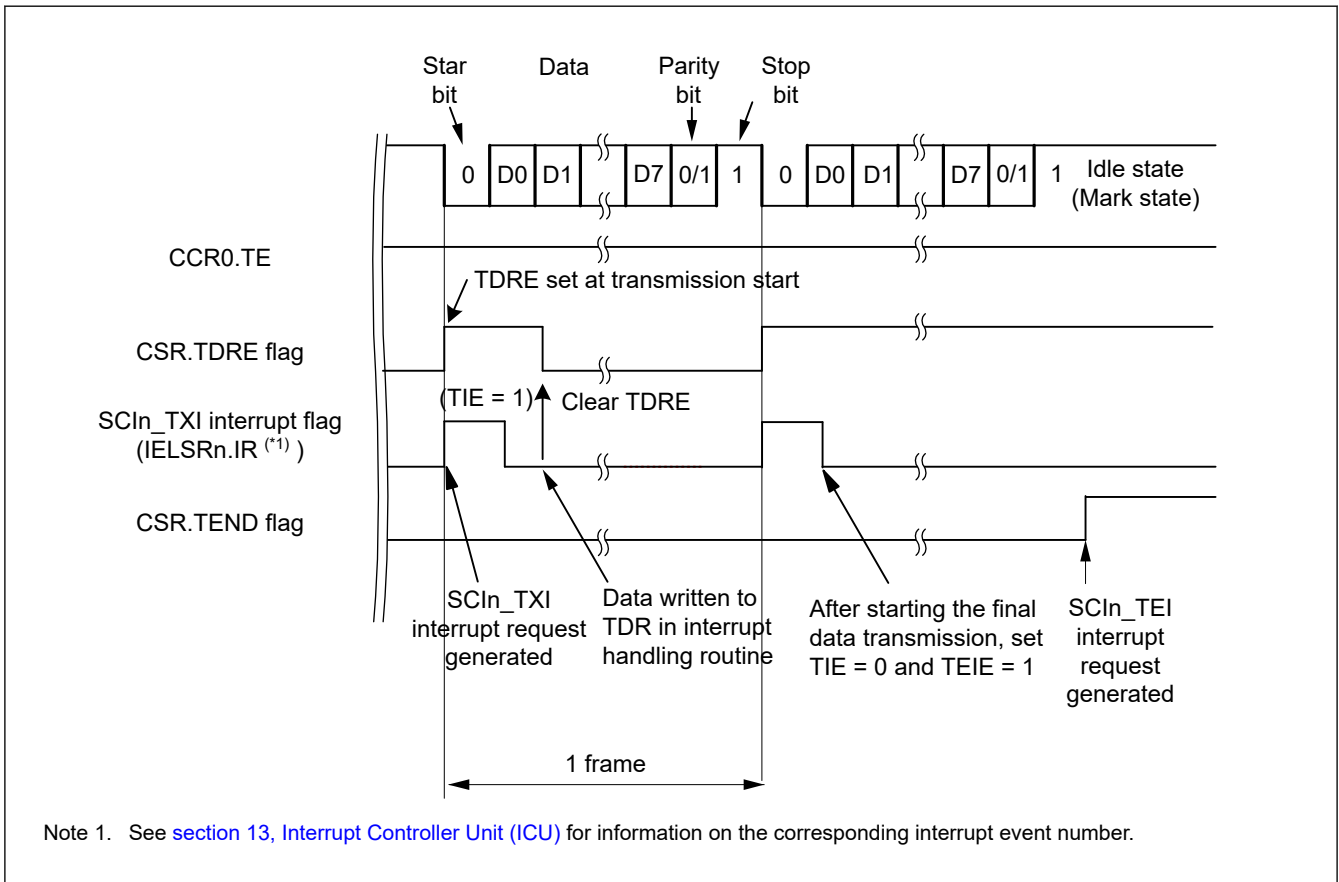
Figure 30.12, Figure 30.13, Figure 30.14 and Figure 30.15 show examples of serial transmission in Asynchronous mode.



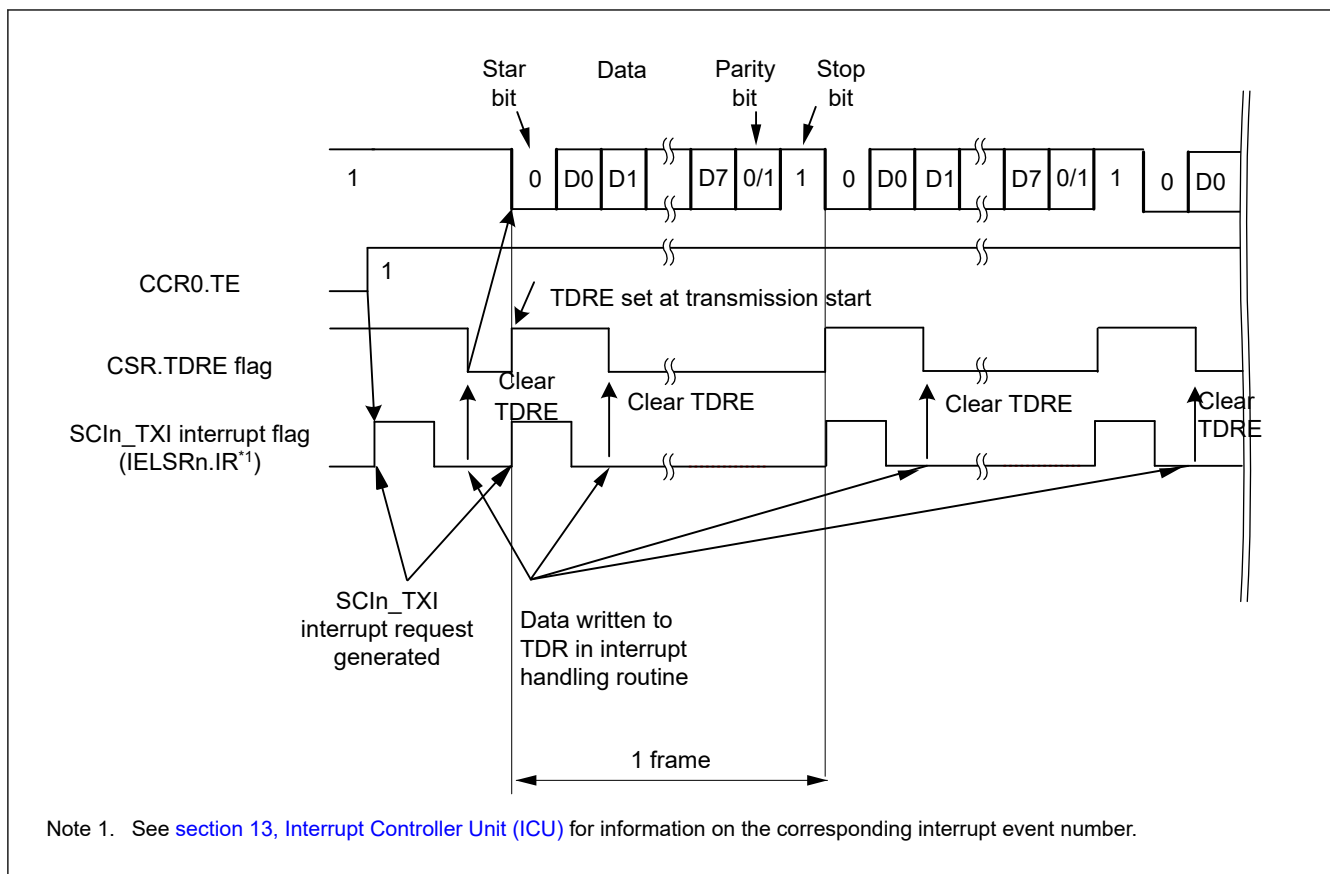
**Figure 30.12 Example operation for serial transmission in Asynchronous mode (1) with 8-bit data, parity bit, 1 stop bit, CTS function not used, and at the beginning of transmission**



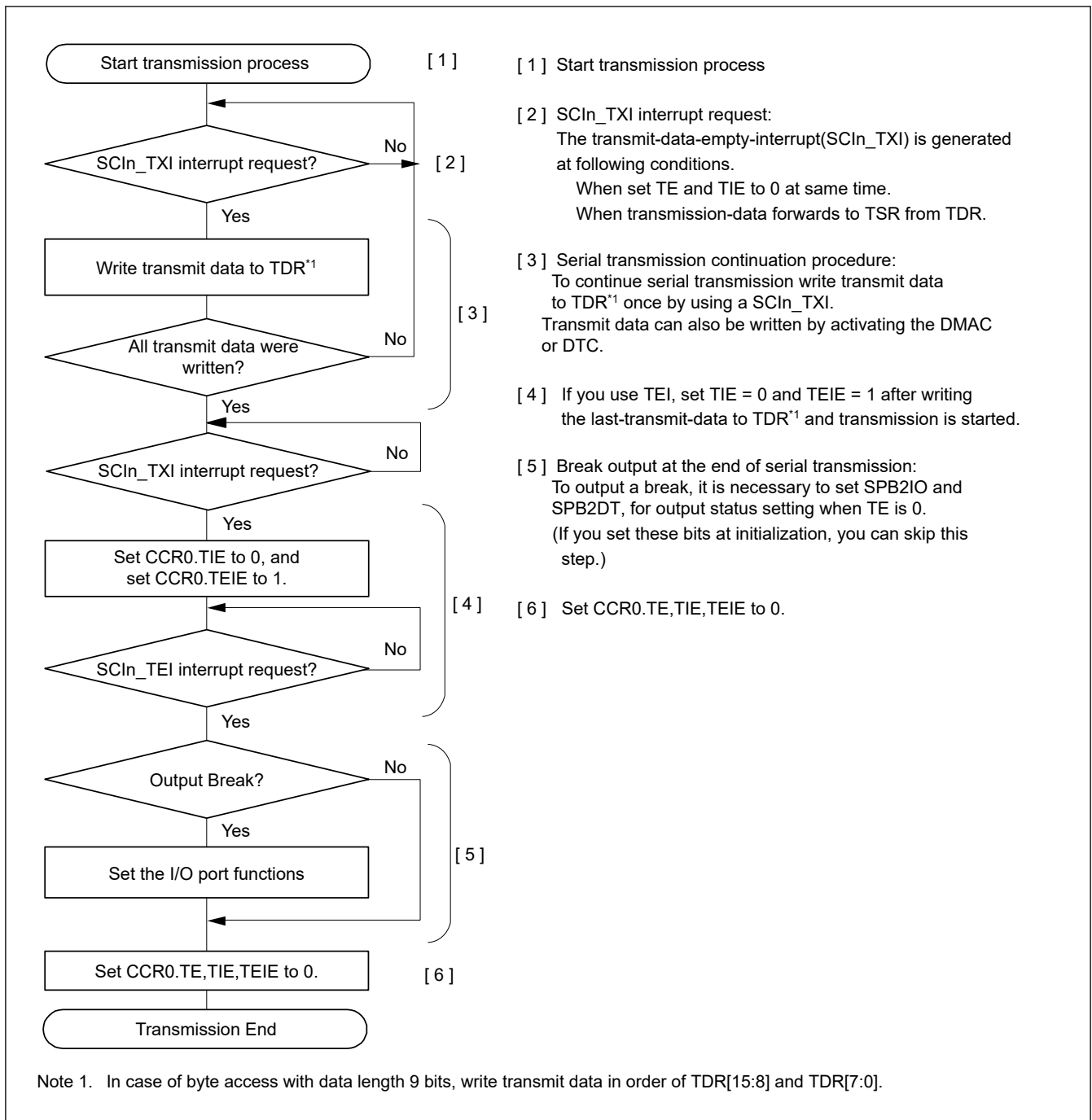
**Figure 30.13** Example operation for serial transmission in Asynchronous mode (2) with 8-bit data, parity bit, one stop bit, CTS function used, and at the beginning of transmission



**Figure 30.14** Example operation for serial transmission in Asynchronous mode (3) with 8-bit data, parity bit, one stop bit, CTS function not used, and from the middle of transmission until transmission completion



**Figure 30.15 Example of Operation for Serial Transmission in Asynchronous Mode (4)(with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)**



**Figure 30.16 Example flow of serial transmission in Asynchronous mode with non-FIFO selected**

**(2) FIFO selected**

Figure 30.17 shows an example of a data format that is written to TDR register in asynchronous mode with FIFO selected.

Data corresponding to the data length is set to TDR[8:0]. Write 0 for unused bits. Write in order from TDR[15:8] to TDR[7:0].



Data Length	Register setting		Transmit data in TDR[15:0]														
	CCR3. CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
7 bit	1	1	-	-	-	-	-	-	MPB T	-	-	TDAT[6:0]					
8 bit	1	0	-	-	-	-	-	-	MPB T	-	TDAT[7:0]						
9 bit	0	Don't Care	-	-	-	-	-	-	MPB T	TDAT[8:0]							

Note: —: Invalid. The write value should be 0.

**Figure 30.17 Data format written to transmit-FIFO(TDR) with FIFO selected**

In serial transmission, the SCI operates as described in this section.

1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn\_TXI interrupt handling routine. The amount of data that can be written to TDR is 16 minus FTSR.T[5:0] bytes. The SCIn\_TXI interrupt request at the beginning of transmission is generated when the CCR0.TE and CCR0.TIE bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CCR1.CTSE bit is set to 0 (CTS function is disabled) or a low level on the CTSn\_RTSn pin causes data transfer from the TDR register to the TSR register. When the amount of transmit data written in TDR is equal to or less than the specified transmit triggering number, CSR.TDRE is set to 1. If the CCR0.TIE bit is 1, an SCIn\_TXI interrupt request is generated. Continuous transmission is possible by writing the next transmit data to TDR in the SCIn\_TXI interrupt handling routine before transmission of the current transmit data is complete. When SCIn\_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 (SCIn\_TXI interrupt requests are disabled) and the CCR0.TEIE bit to 1 (an SCIn\_TEI interrupt request is enabled) after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn\_TXI requests.
3. Data is sent from the TXDn pin in the following order:
  - Start bit
  - Transmit data
  - Parity bit or multi-processor bit (can be omitted depending on the format)
  - Stop bit
4. On output of the stop bit, the SCI checks whether non-transmitted data remains in the TDR register.
5. When data is set to transmit-FIFO (TDR), setting the CCR1.CTSE bit to 0 (CTS function is disabled) or a low level input on the CTSn\_RTSn pin causes transfer of the next transmit data from TDR to TSR and transmission of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set in transmit-FIFO (TDR), the TEND flag in CSR is set to 1, the stop bit is sent, and the mark state is entered in which 1 is output. If the CCR0.TEIE bit is 1, an SCIn\_TEI interrupt request is generated.

Figure 30.18 shows an example flow of serial transmission in Asynchronous mode with FIFO selected.

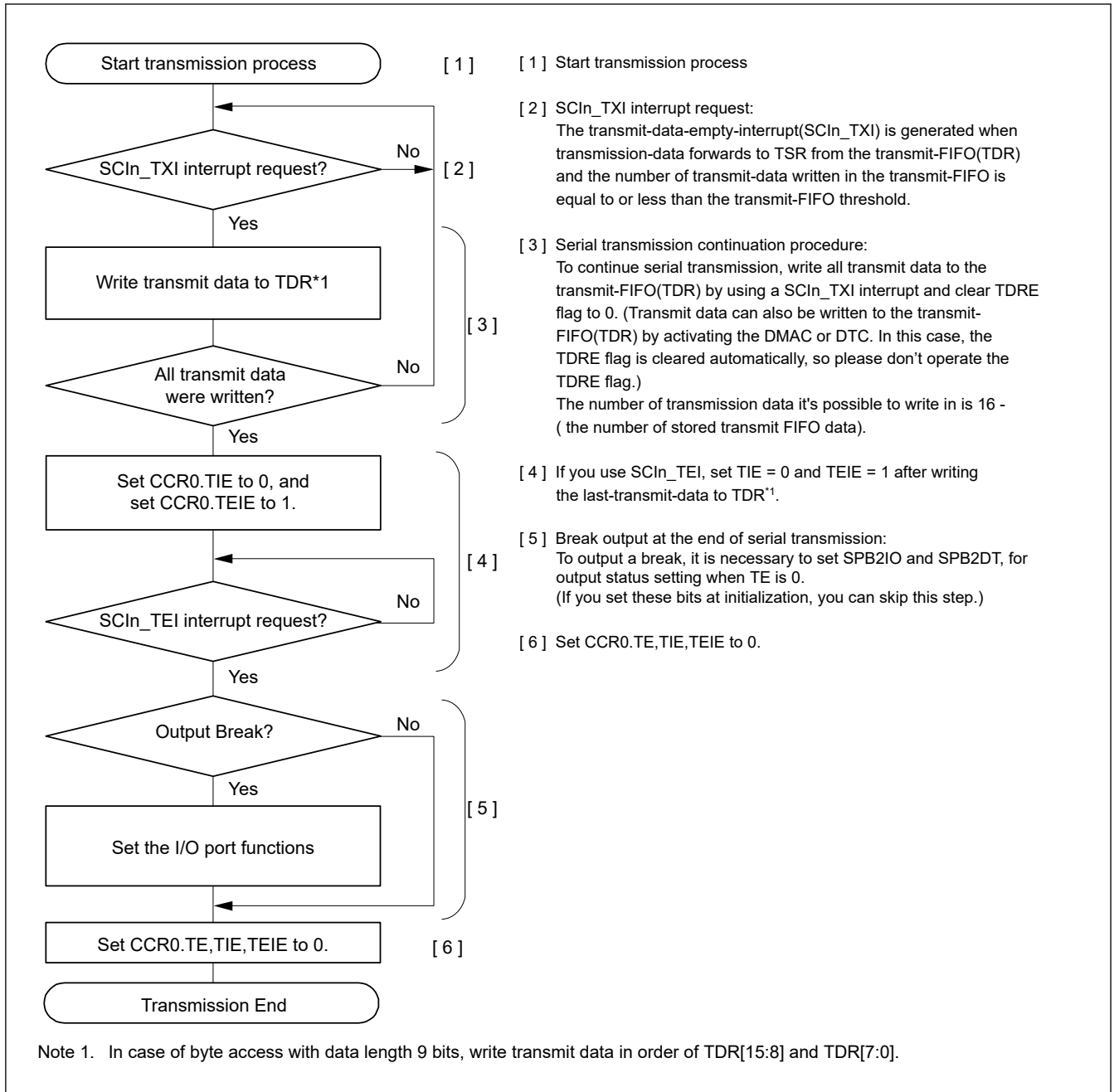


Figure 30.18 Example flow of serial transmission in Asynchronous mode with FIFO selected

### 30.3.9 Serial Data Reception in Asynchronous Mode

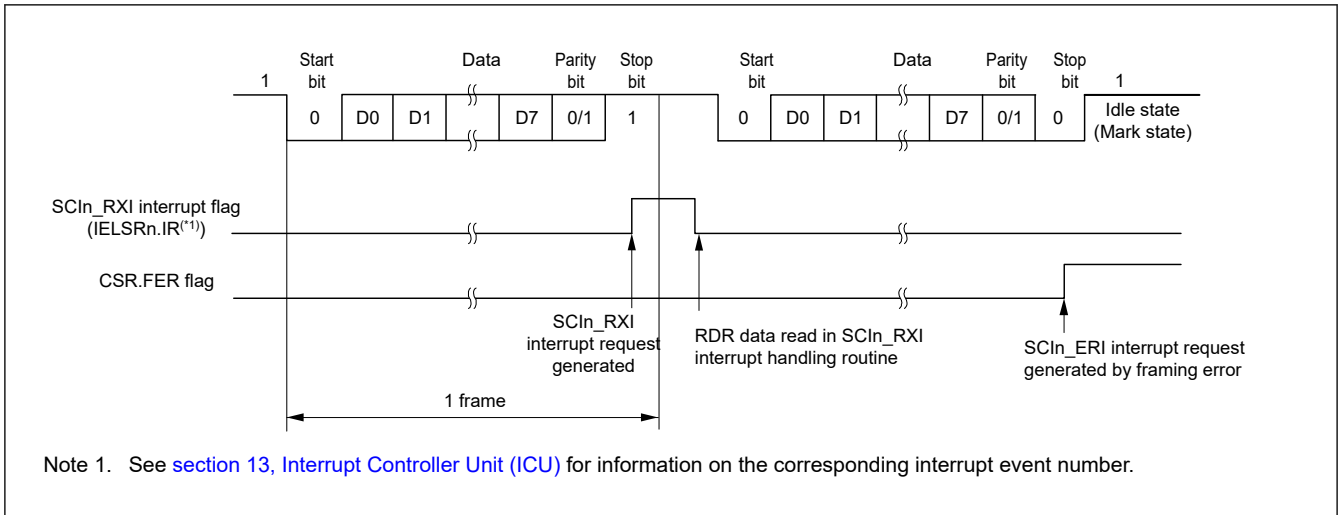
#### (1) Non-FIFO selected

Figure 30.19 and Figure 30.20 show an example of the operation for serial data reception in Asynchronous mode.

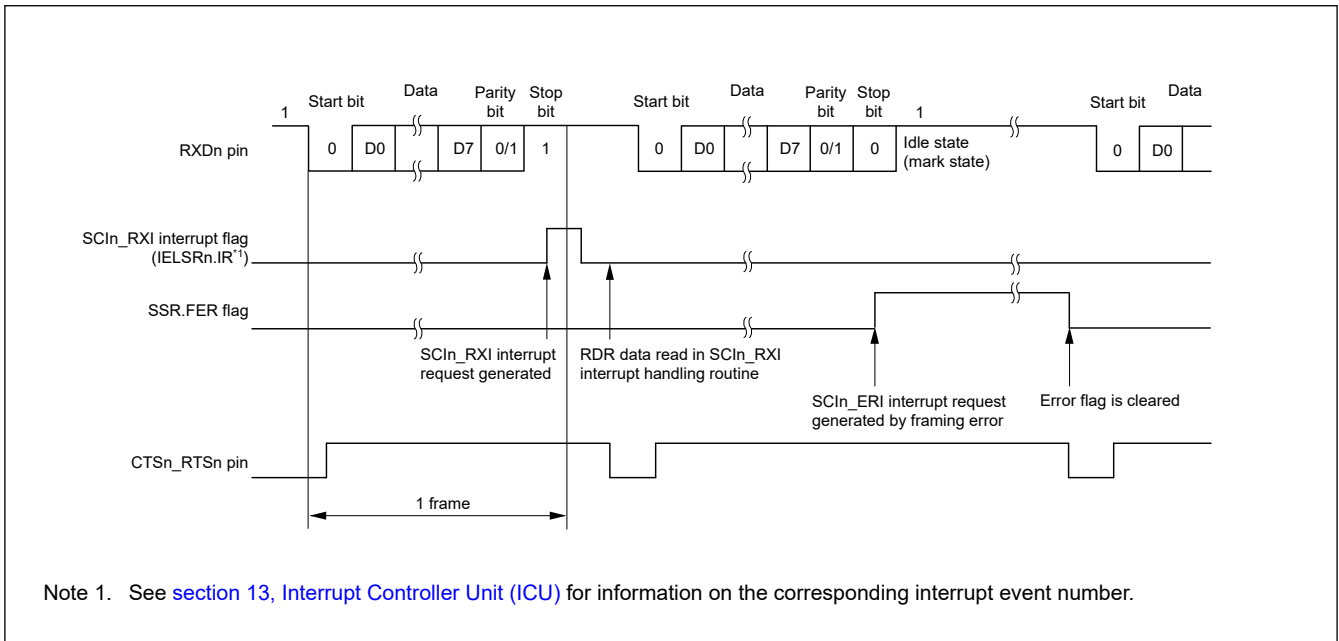
In serial data reception, the SCI operates as follows:

1. When the value of the CCR0.RE bit becomes 1, the output signal on the CTSn\_RTSn pin goes low.
2. The SCI monitors the communications line and when it detects a start bit, the SCI performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn\_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. If a parity error is detected, the CSR.PER flag is set to 1 and receive data is transferred to the RDR register. If the SCR.RIE bit is 1, an SCIn\_ERI interrupt request is generated.

5. If a framing error is detected, the CSR.FER flag is set to 1 and receive data is transferred to the RDR register. If the CCR0.RIE bit is 1, an SCIn\_ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to the RDR register. If the CCR0.RIE bit is 1, an SCIn\_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in the SCIn\_RXI interrupt handling routine before reception of the next receive data is complete. Reading the received data that was transferred to the RDR register causes the CTSn\_RTSn pin to output low.



**Figure 30.19 Example of SCI operation for serial reception in Asynchronous mode (1) when the RTS function is not used, and with 8-bit data, parity bit, and 1 stop bit**



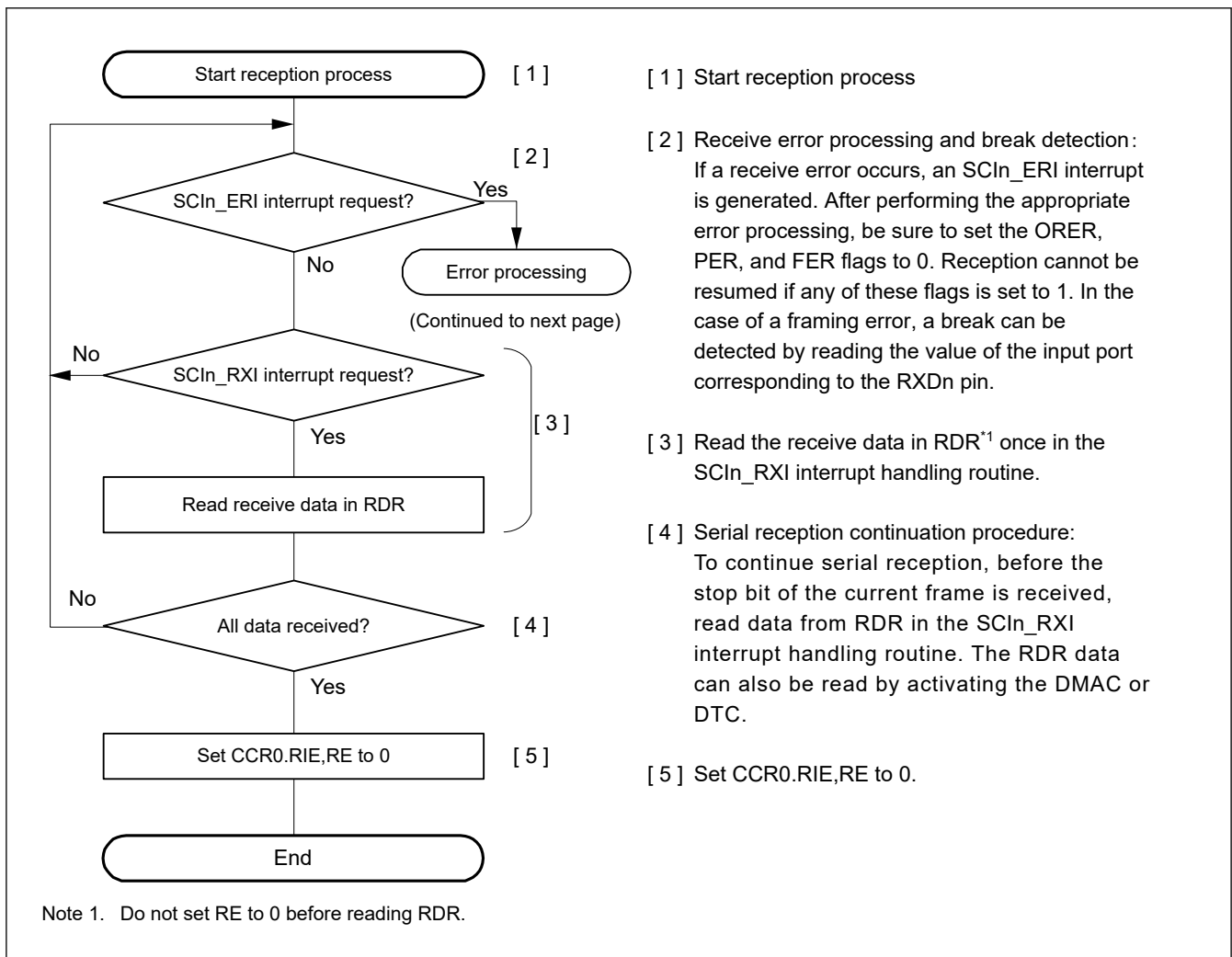
**Figure 30.20 Example of SCI operation for serial reception in Asynchronous mode (2) when RTS function is used, and with 8-bit data, parity bit, and 1 stop bit**

[Table 30.31](#) lists the states of the flags in the CSR status register and receive data handling when a receive error is detected. If a receive error is detected, an SCIn\_ERI interrupt request is generated but an SCIn\_RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the OREr, FER, and PER bits to 0 before resuming reception. In addition, be sure to read the RDR register during overrun error processing. When a reception is forced to terminate by setting the CCR0.RE bit to 0 during operation, read the RDR register because received data that is not yet read might be left in the RDR.

[Figure 30.21](#) and [Figure 30.22](#) show example flows of serial data reception.

**Table 30.31 Flags in CSR Status Register and receive data handling**

Flags in the CSR Status Register			Receive data	Receive error type
ORER	FER	PER		
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error



**Figure 30.21 Example flow of serial reception in Asynchronous mode with non-FIFO selected and Address Matching Disabled (1)**

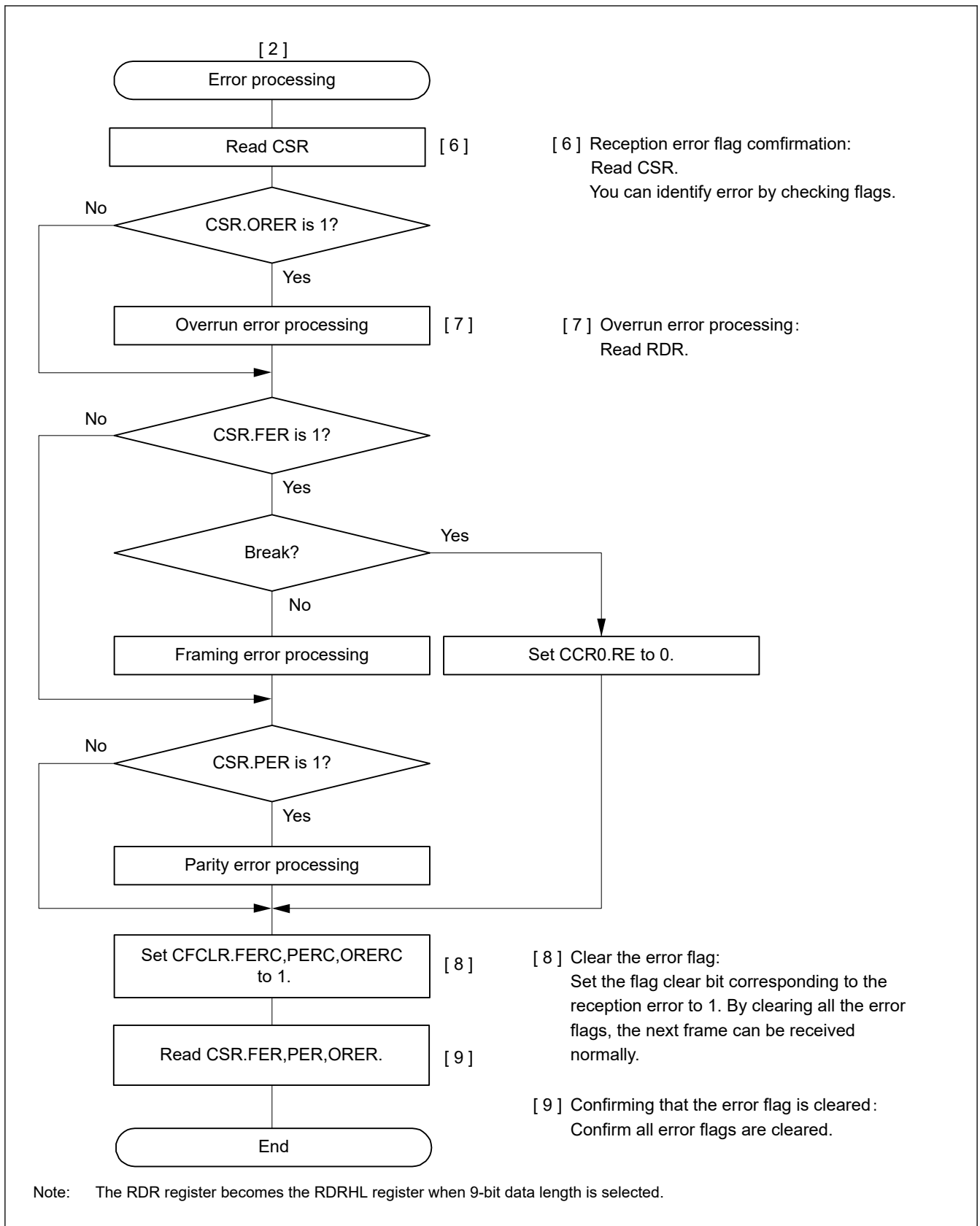


Figure 30.22 Example flow of serial reception in Asynchronous mode with non-FIFO selected and Address Matching Disabled (2)

(2) FIFO selected

Figure 30.23 shows an example of a data format that is written to Receive-FIFO(RDR) register in Asynchronous mode.

In Asynchronous mode, 0 is written to the MPB bit in the RDR register. Data that corresponds to the data length is written to RDR. Unused bits are written as 0. If software reads RDR, the SCI updates FER, PER, and receive data (RDAT[8:0]) in the RDR register with the next data. The flags ORER, and DR in the RDR register always reflect the associated flags in the CSR register.

Data Length	Register Setting		Receive flag in RDR[31:0], MPB, RDAT[8:0]																		
	CCR3.CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0			
7bit	1	1	-	-	-	FFER	FPER	DR	MPB	0	0	RDAT[6:0]									
8bit	1	0	-	-	-	FFER	FPER	DR	MPB	0	RDAT[7:0]										
9bit	0	Don't care	-	-	-	FFER	FPER	DR	MPB	RDAT[8:0]											
		CCR3.CHR[1:0]	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16			
7bit	1	1	-	-	-	FER	PER	-	-	ORER	-	-	-	-	-	-	-	-			
8bit	1	0	-	-	-	FER	PER	-	-	ORER	-	-	-	-	-	-	-	-			
9bit	0	Don't care	-	-	-	FER	PER	-	-	ORER	-	-	-	-	-	-	-	-			

Note: 0 is always read from the MPB flag (RDR [9] bit).  
 When a 7-bit data length is selected, 0 is read from the RDAT [8:7] bits.  
 When 8-bit data length is selected, 0 is read from the RDAT [8] bit.

Figure 30.23 Data format stored in receive-FIFO(RDR) with FIFO selected

Table 30.32 lists the states of the flags in CSR status register and receive data handling when a receive error is detected with FIFO selected. Figure 30.24 and Figure 30.25 show samples of flowcharts for serial data reception with FIFO selected. In serial data reception, the SCI operates as follows:

1. When the value of the CCR0.RE bit becomes 1, the output signal on the CTSn\_RTSn pin goes low.
2. The SCI monitors the communications line and, when it detects a start bit, the SCI performs internal synchronization, stores receive data in the RSR register.
3. If an overrun error occurs during normal communications, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn\_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. If a parity error is detected, the PER flag and receive data are transferred to the RDR register. If the CCR0.RIE bit is set to 1, an SCIn\_ERI interrupt request is generated.
5. If a framing error is detected, the FER flag and receive data are transferred to the RDR register. If the CCR0.RIE bit is set to 1, an SCIn\_ERI interrupt request is generated.
6. After a framing error is detected and when SCI detects that the continuous receive data is zero for one frame, reception stops.
7. When the amount of data stored in the RDR register falls below the specified receive triggering number, and the next data is not received after 15 etus from the last stop bit in Asynchronous mode, the FRSR.DR flag is set to 1. When the CCR0.RIE bit is 1 and the FCR.DRES bit is 0, the SCI generates an SCIn\_RXI interrupt request. When the FCR.DRES bit is 1, SCI generates an SCIn\_ERI interrupt request.
8. When reception finishes successfully, receive data is transferred to the RDR register. The RDRF bit is set to 1 when the amount of receive data written to RDR is equal to or greater than the specified receive triggering number. If the CCR0.RIE bit is 1, an SCIn\_RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in the SCIn\_RXI interrupt handling routine, before an overrun error occurs. If the received data that is transferred to RDR is less than the RTS trigger number, the CTSn\_RTSn pin outputs low.

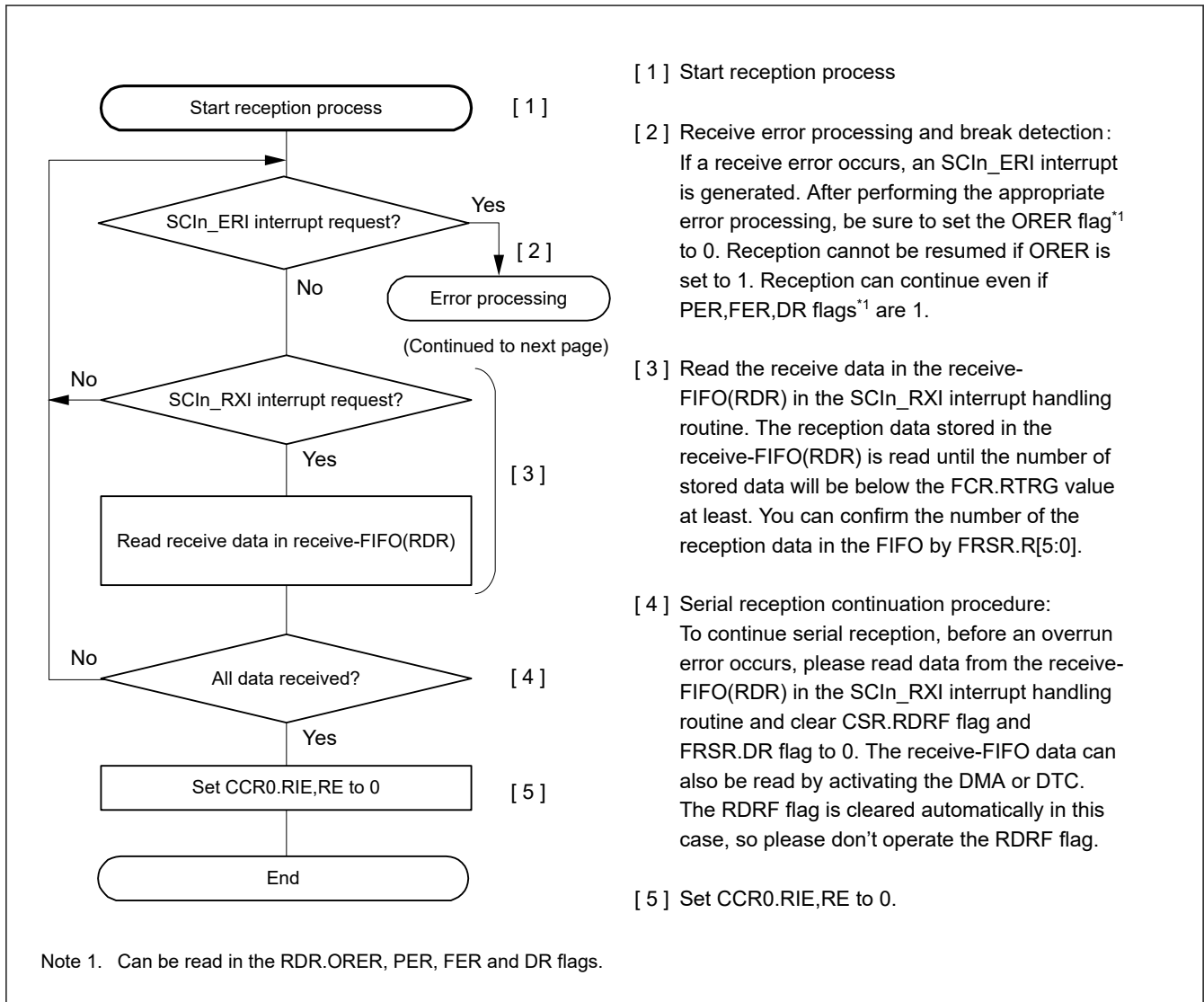
Table 30.32 Flags in the CSR Status Register and Receive Data Handling (FIFO selected) (1 of 2)

CSR value			Receive-FIFO (RDR)	Receive Error Type
ORER	FER*1	PER*1	RDAT[8:0]	
1	0	0	Lost	Overrun error
0	1	0	Transferred RDR	Framing error
0	0	1	Transferred RDR	Parity error

**Table 30.32 Flags in the CSR Status Register and Receive Data Handling (FIFO selected) (2 of 2)**

CSR value			Receive-FIFO (RDR)	Receive Error Type
ORER	FER <sup>*1</sup>	PER <sup>*1</sup>	RDAT[8:0]	
1	1	0	Lost	Overrun error + Framing error
1	0	1	Lost	Overrun error + Parity error
0	1	1	Transferred RDR	Framing error + Parity error
1	1	1	Lost	Overrun error + Framing error + Parity error

Note 1. This flag indicates whether there is an error in received data when reception is completed.



**Figure 30.24 Example flow of serial reception in Asynchronous mode with FIFO selected and Address Matching Enabled (1)**

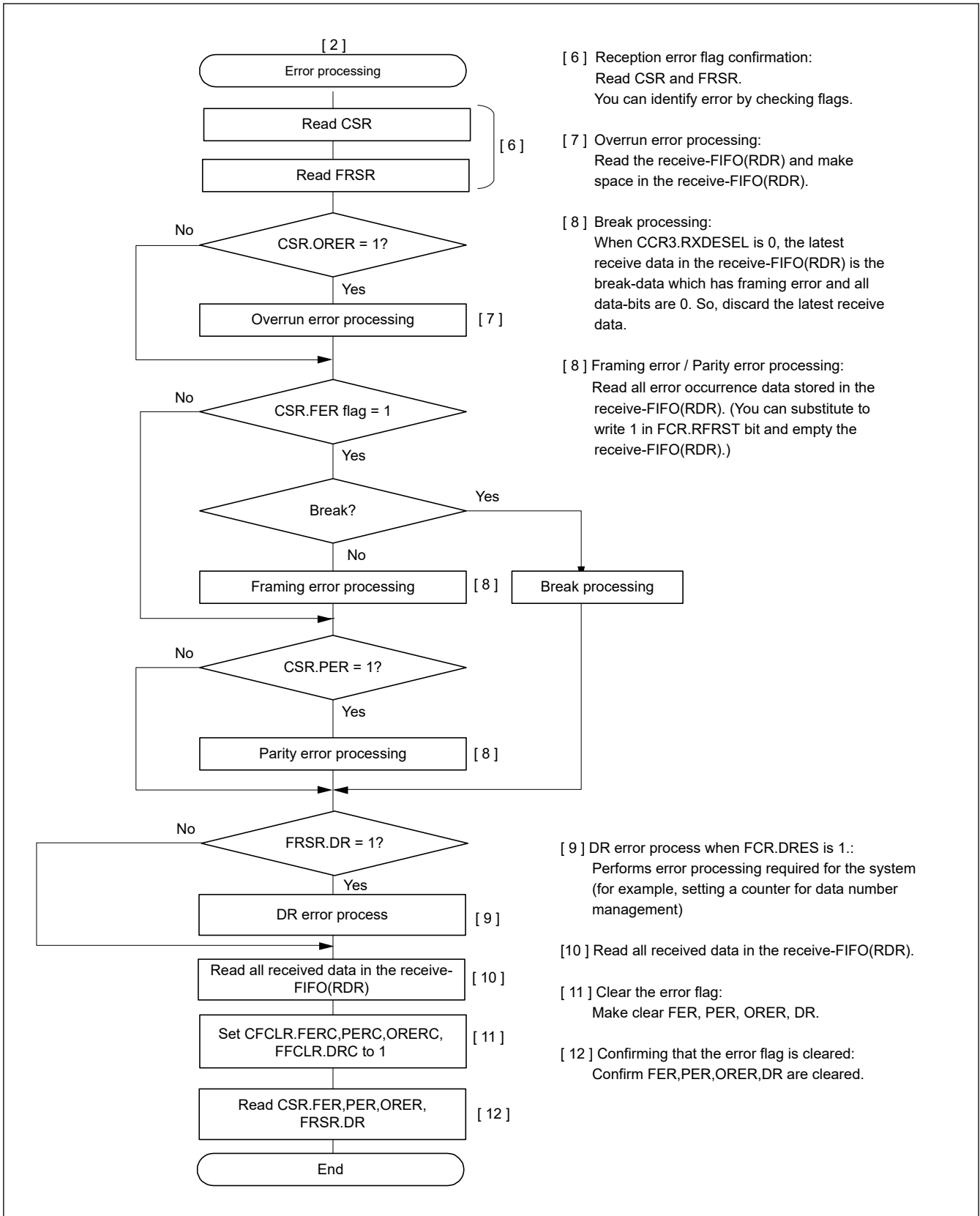


Figure 30.25 Example flow of serial reception in Asynchronous mode with FIFO selected Address Matching Disabled (2)



### 30.3.10 The function of adjust receive sampling timing (Asynchronous Mode)

When there is the difference between the rising transfer time and the falling transfer time through a photo coupler, the receive sampling timing at middle of bit affects the reception margin. In this case, the receive sampling timing is able to adjust from the middle of bit to the optimum timing by using this function.

The receive sampling timing is adjusted from the middle of bit by following formula. The adjustable direction is set by CCR4.AJD. When adjusting backward (CCR4.AJD = 0), substitute AJD = +1 and substitute AJD = -1 when adjusting forward (CCR4.AJD = 1).

$$\text{Adjusted sampling timing} = \text{the middle of bit} + \text{AJD} \times (\text{base clock} \times \text{the setting value of CCR4.AST}[2:0])$$

The setting timing is limited by base clock cycles per 1 bit. For details, see [Table 30.33](#).

An overview of reception operation of the communication through a photo coupler with this function is shown in [Figure 30.26](#), [Figure 30.27](#) and [Figure 30.28](#), the explanation of operation with this function is shown in [Figure 30.29](#).

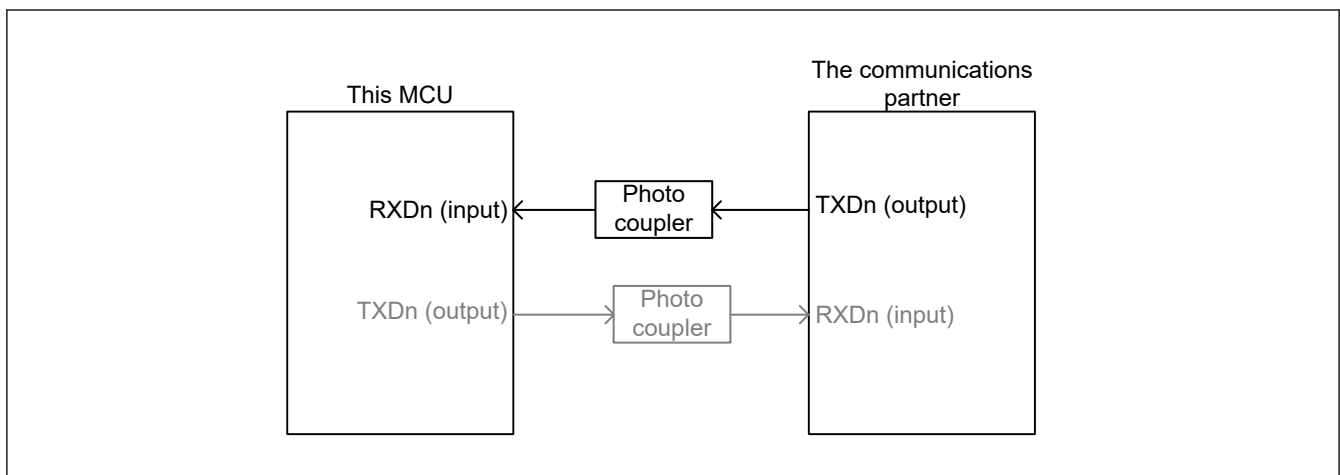
Do not use this function when there is no difference between the rising transfer time and the falling transfer time, because there is a possibility of deteriorating the reception margin.

**Table 30.33 The acceptable value of setting register (Asynchronous mode using internal clock)**

CCR2.ABCSE2	CCR2.ABCSE	CCR2.ABCS	The number of base cycles/1bit	The acceptable value	
				CCR4.AJD	CCR4.AST*1
1	0	x	4	0	000~011
				1	(Note1)
0	1	x	6	0	000~010
				1	(Note1)
0	0	1	8	0	000~011
				1	(Note1)
0	0	0	16	0	000~111
				1	

Note: x: Don't care

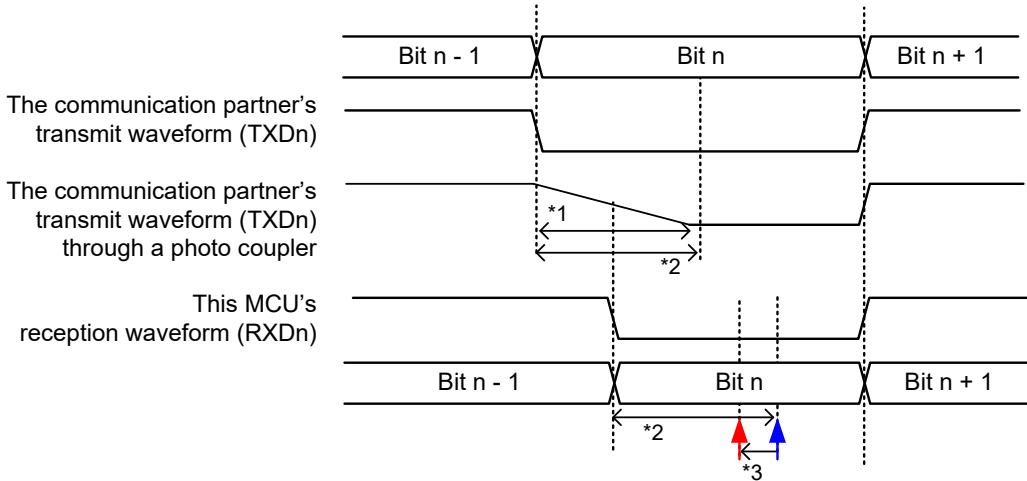
Note 1. When the value of CCR4.AST exceeds the acceptable value, sampling is done at default timing. (Adjustment of sampling is not done.)



**Figure 30.26 block diagram image of the reception through a photo couple**

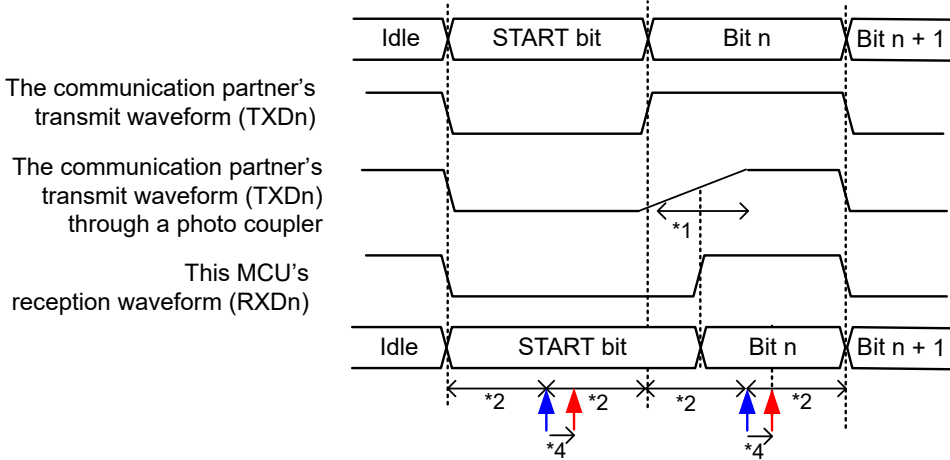
(a) In the case of the falling transfer time >> rising transfer time

The falling edge of reception waveform is made dull like following chart.  
 In this case, you can sampling at the middle of bit if you adjust the receive sampling timing to forward (AJD = 1).



(b) In the case of the falling transfer time << rising transfer time

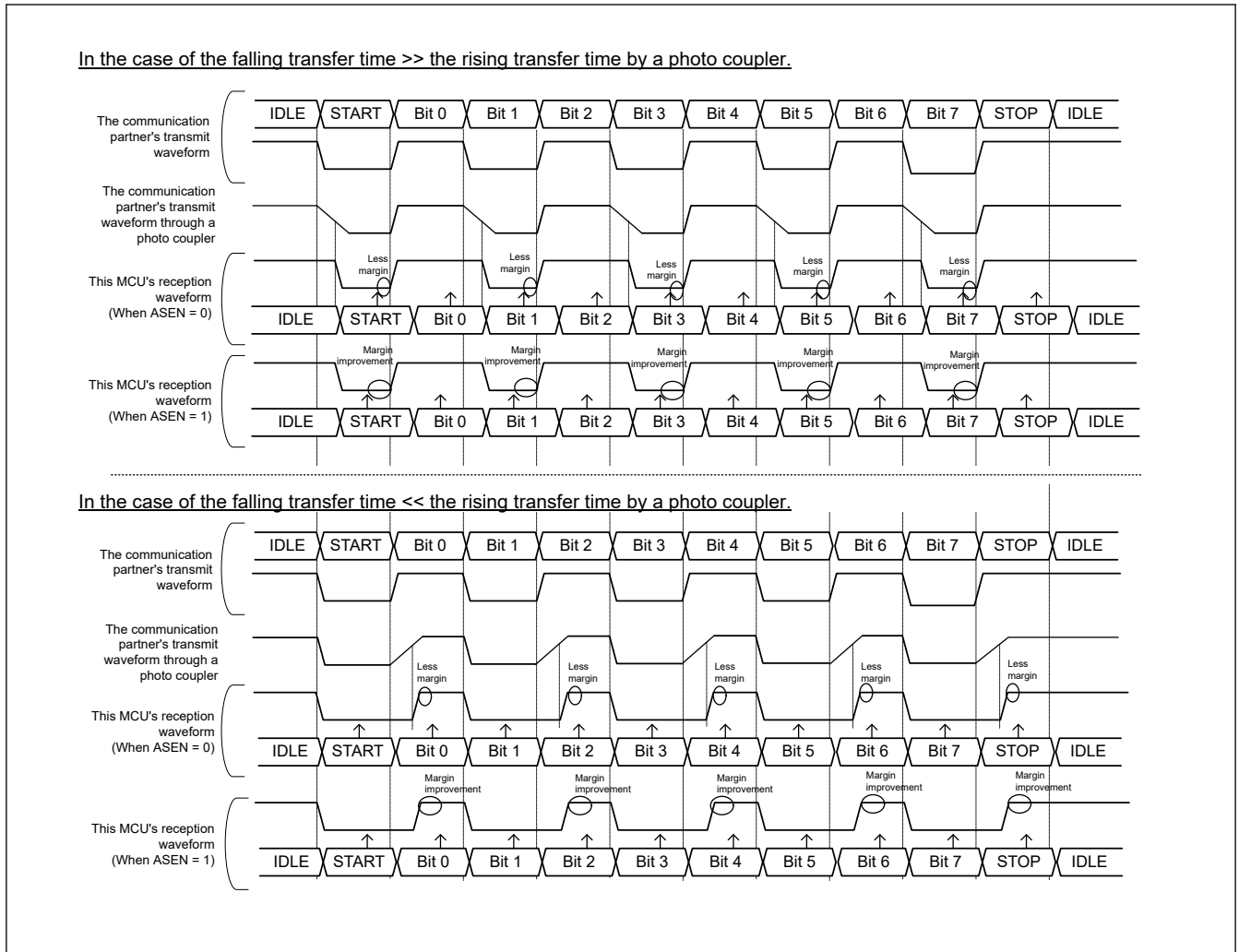
The rising edge of reception waveform is made dull like following chart.  
 So, the reception margin of communications partner will be bad. In this case, you can improve the reception margin if you adjust the receive sampling timing to back.



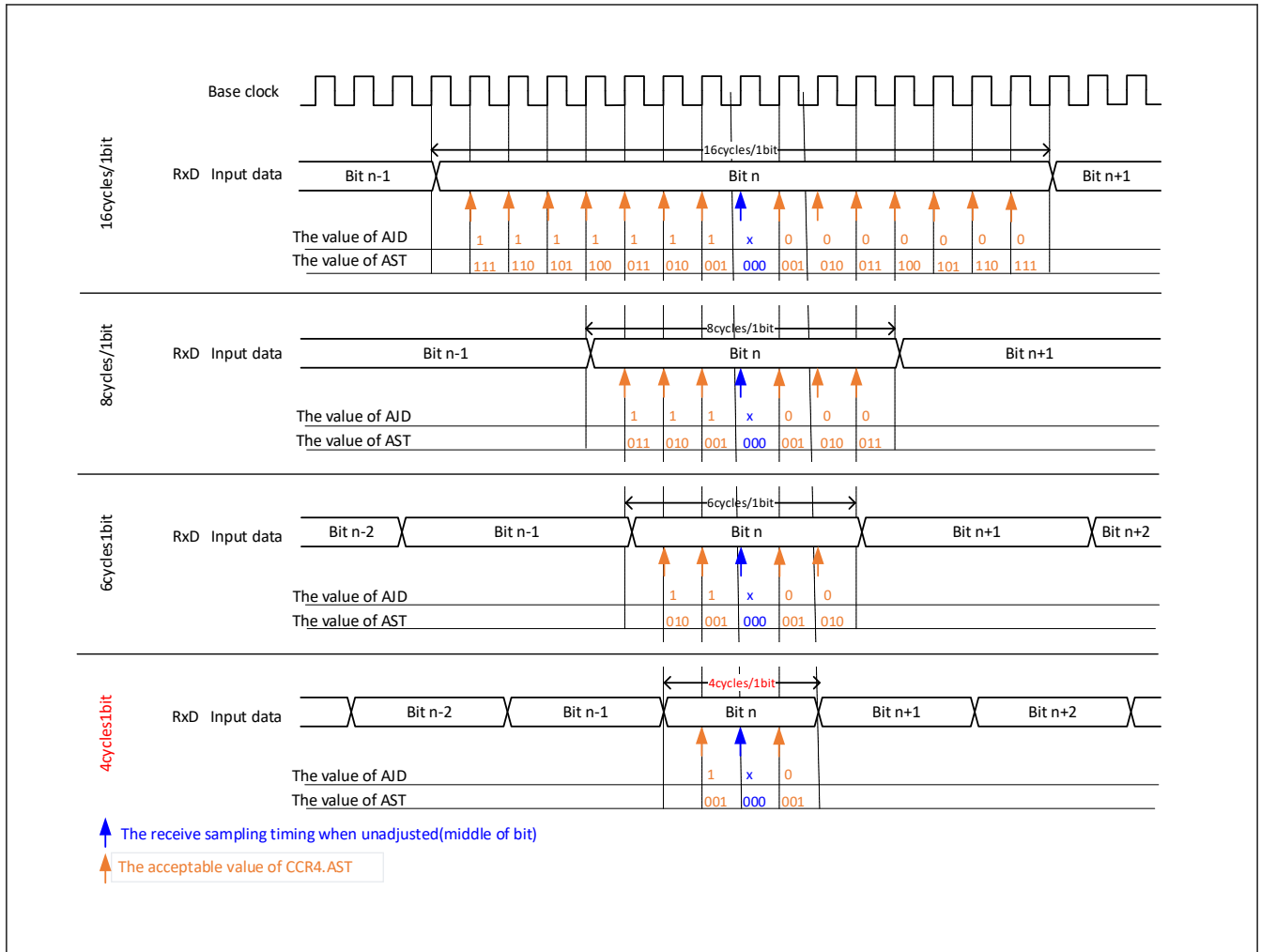
- ▲ The receive sampling timing when unadjusted (middle of bit)
- ▲ The adjusted receive sampling timing

- Note: This waveform shows the operation image of adjustment in reception sampling timing.
- Note 1. The dull period by a photo coupler
  - Note 2. Bit center timing at set communication rate
  - Note 3. When CCR4.AJD is 1, the receive sampling timing is shifted to forward by the setting value of CCR4.AST[2:0].
  - Note 4. When CCR4.AJD is 0, the Receive sampling timing is shifted to backward by the setting value of CCR4.AST[2:0].

**Figure 30.27 Overview of reception operation of the communication through a photo coupler**



**Figure 30.28 Example of improvement in reception margin by the reception sampling timing adjustment function**



**Figure 30.29 Overview of the adjustment operation for the reception sampling timing (Asynchronous mode using internal clock)**

### 30.3.11 The function of adjust transmit timing (Asynchronous Mode)

In communication via a photo coupler or the like, when either the rising or falling transition time of the TXDn output signal is long, then a communication partner receive dulled waveform. In this case, the reception margin may be affected.

In these cases, make a communication partner to be sampling at middle of bit using the function of adjust transmit timing.

When CCR4.ATEN is 1, this function can adjust the edge timing at the timing calculated by the following formula for the edge set with CCR4.AET.

$$\text{The adjustment edge timing} = \text{the base clock} \times \text{CCR4.ATT}[2:0]$$

In addition, the upper limit of the adjustment edge timing is limited by setting the base clock cycles. For details, see [Table 30.34](#).

A transmission movement image figure of the communication through a photo coupler with this function is shown in [Figure 30.30](#), [Figure 30.31](#) and [Figure 30.32](#), the overview of operation with this function is shown in [Figure 30.33](#) and [Figure 30.34](#).

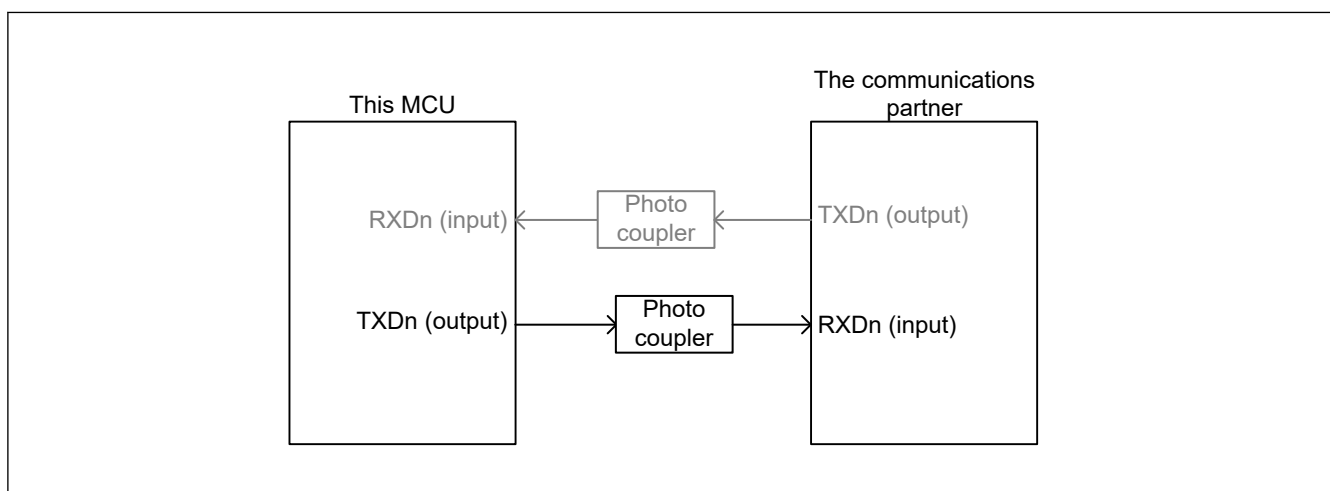
Do not use this function when there is not the difference between the rising transfer time and the falling transfer time, there is a possibility of deteriorating the reception margin of a communication partner.

**Table 30.34** The acceptable value of CCR4.AET and CCR4.ATT (Asynchronous mode using internal clock)

ABCSE2	ABCSE	ABCS	The number of base clock cycles/1bit	The acceptable value	
				AET	ATT [2:0]
1	0	x	4	0 1	000~011
0	1	x	6	0 1	000~101
0	0	1	8	0 1	000~111
0	0	0	16	0 1	000~111

Note: x: Don't care

Note: When the value of ACTR.AET/ATT is out of the acceptable value, this SCI module doesn't adjust transmit timing.



**Figure 30.30** block diagram image of the transmission through a photo coupler

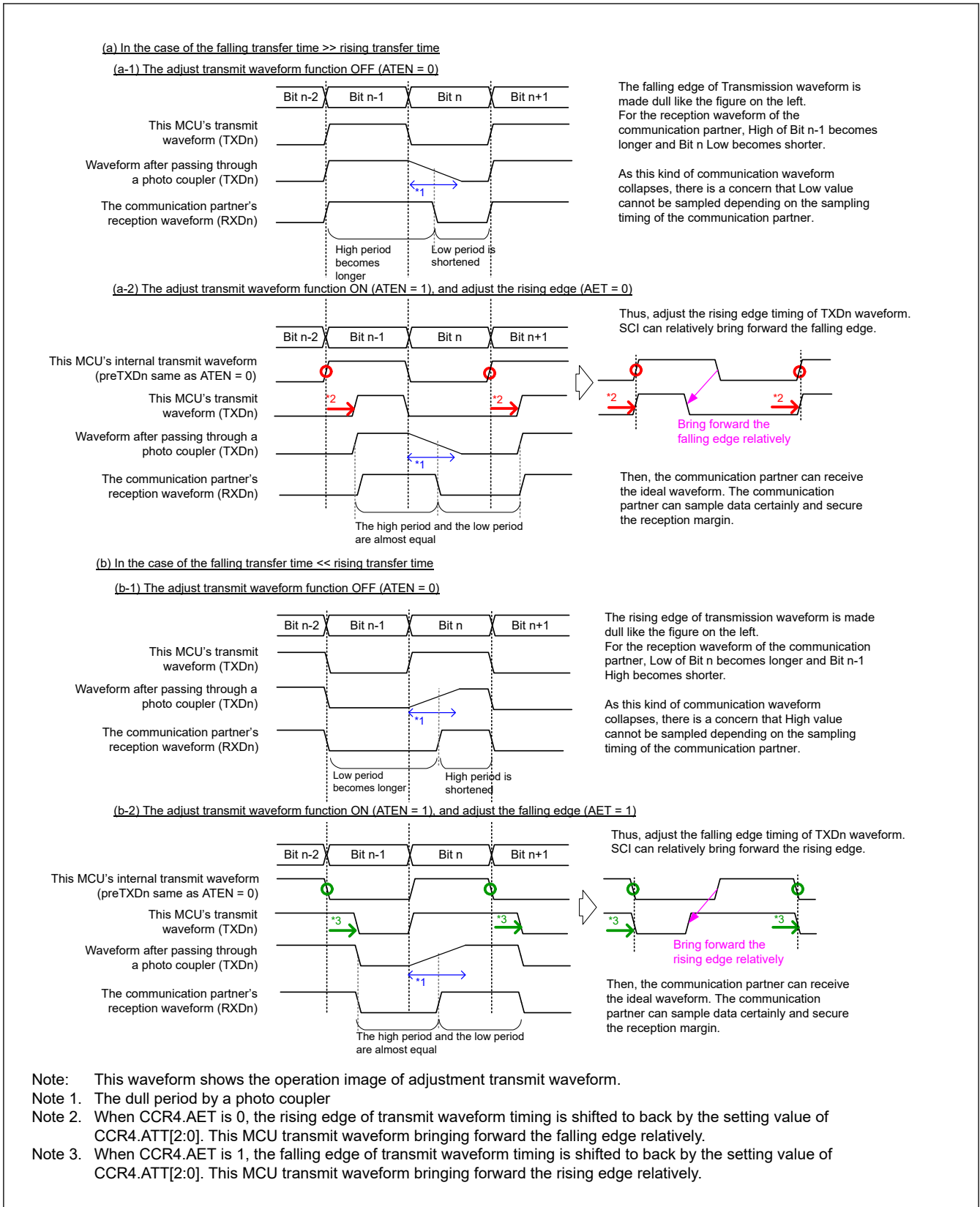


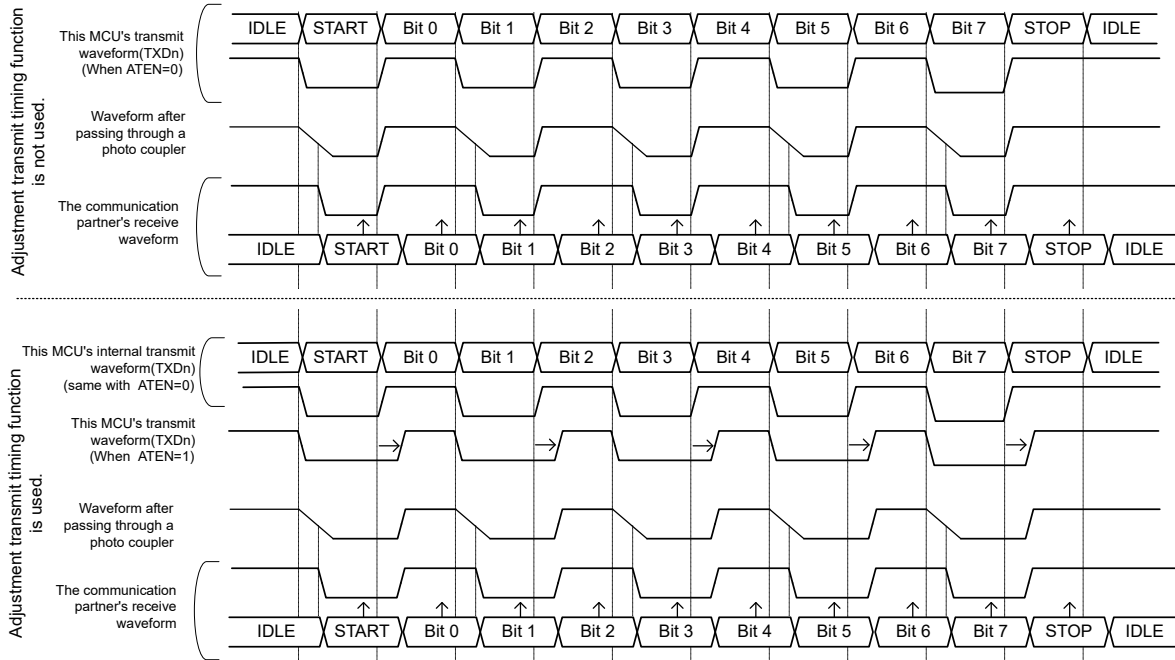
Figure 30.31 The overview of transmission operation in the communication through a photo coupler

The explanation of transmit waveforms of the communication through a photo coupler using adjust transmit timing function

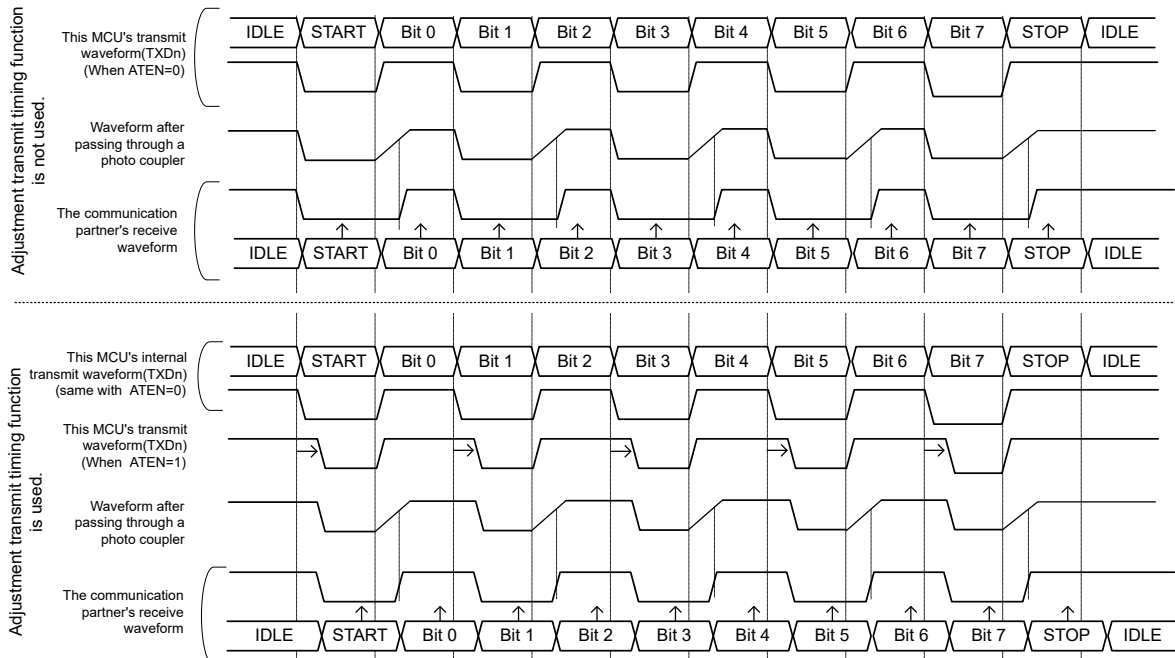
When using the transmission timing adjustment function, adjust the edge timing of the transmission waveform and correct the reception waveform of the communication partner

The following example is 8 bit long data.

(a) In the case of the falling edge transfer time >> the rising transfer time

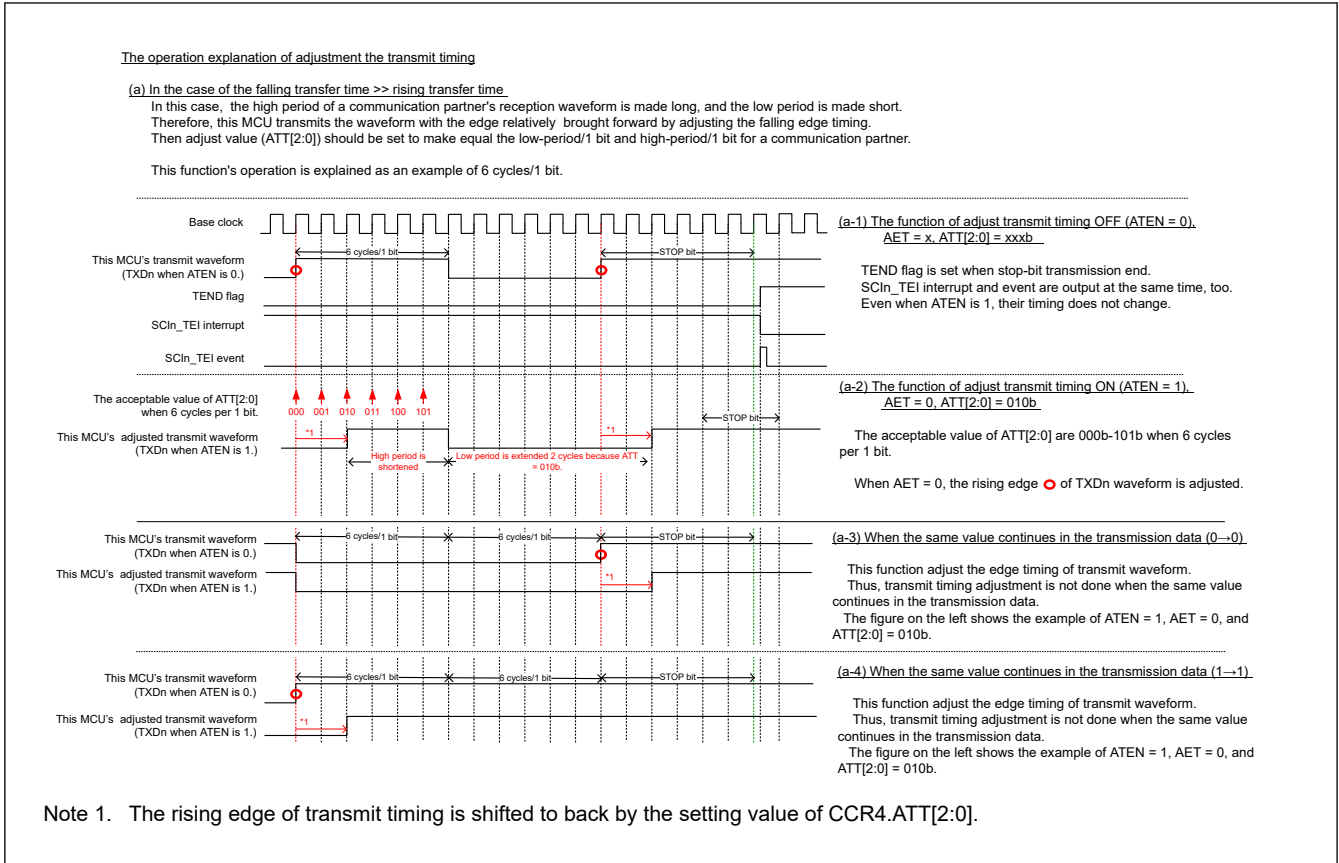


(b) In the case of the falling edge transfer time << the rising transfer time

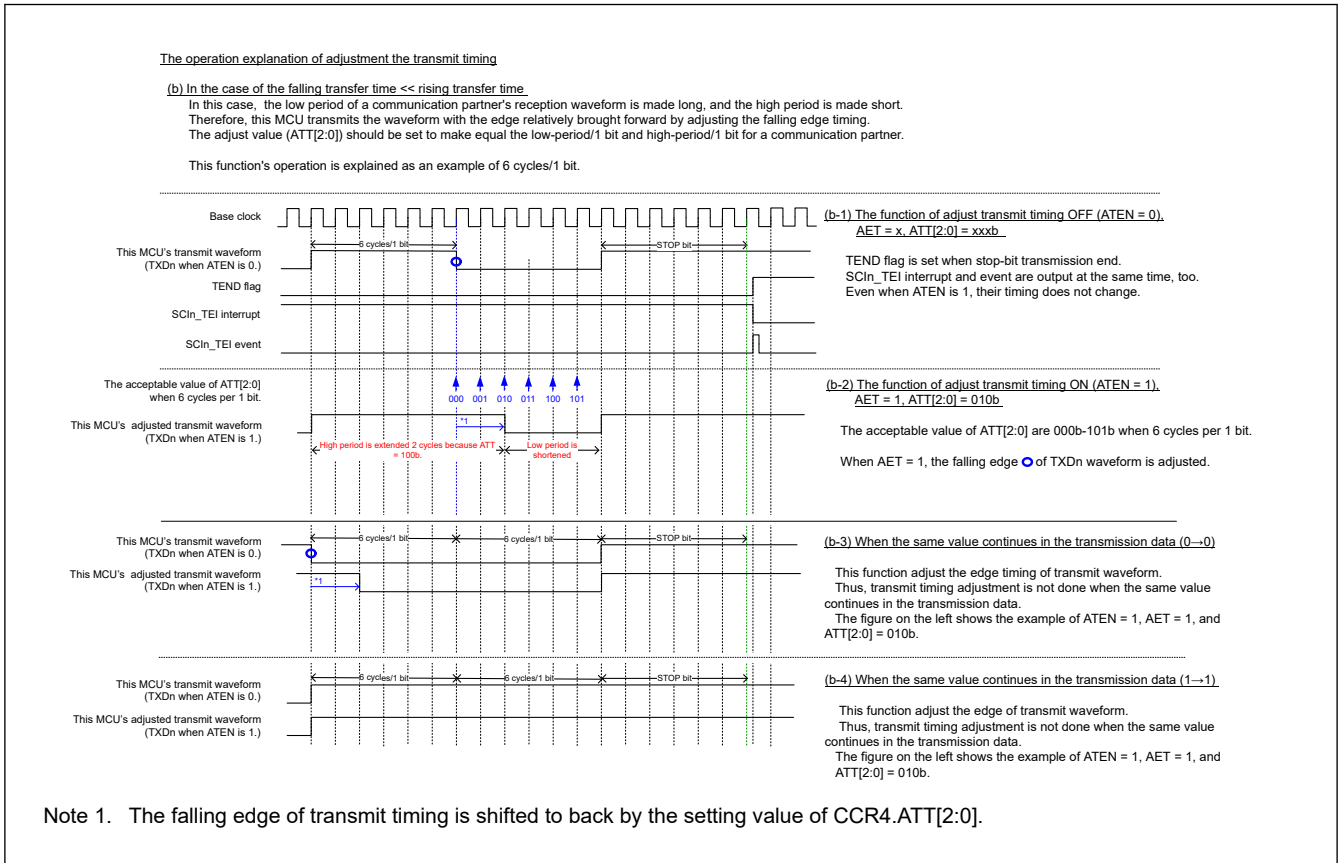


→ : The adjustment edge timing using this function    ↑ : A communication partner's sampling timing

Figure 30.32 The explanation for the transmit waveform through a photo coupler



**Figure 30.33 The adjustment operation explanation for the transmit timing when AET is 0**



**Figure 30.34 The adjustment operation explanation for the transmit timing when AET is 1**



### 30.4 Multi-Processor Communication Function

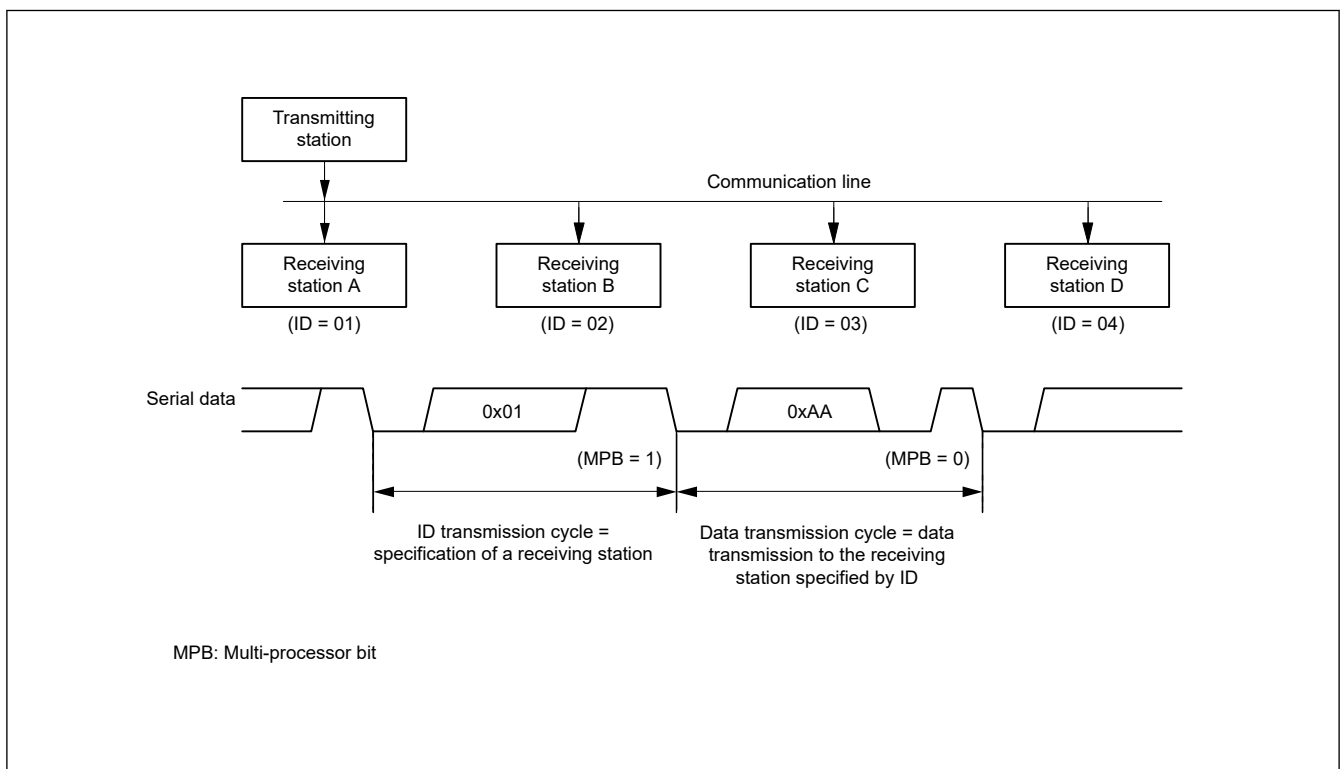
The multi-processor communication function enables the SCI to transmit and receive data between multiple processors by sharing an asynchronous serial communication line that has an added multi-processor bit. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station.

The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle:

- When the multi-processor bit is set to 1, the transmission cycle is the ID transmission cycle
- When the multi-processor bit is set to 0, the transmission cycle is the data transmission cycle

Figure 30.35 shows an example of communication between processors using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits communication data in which the multi-processor bit set to 0 is added to the transmit data. After receiving communication data with the multi-processor bit set to 1, the receiving station compares the received ID with the ID of the receiving station itself. If the two match, the receiving station receives communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until it receives data in which the multi-processor bit is set to 1.

RTS control cannot be used at the time of multi-processor communication function use because this is a function corresponding to one-to-many communications.



**Figure 30.35 Example of communication using multi-processor format with transmission of data 0xAA to receiving station A**

#### (1) Non-FIFO selected

To support this function, the SCI provides the CCR0.MPIE bit. When the MPIE bit is set to 1, the following operations are disabled until the reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the RDR register
- Detection of a receive error
- Setting of the respective RDRF, ORER, and FER status flags in the CSR register

When the SCI receives a character in which the multi-processor bit is set to 1, the RDR.MPB bit is set to 1 and the CCR0.MPIE bit is automatically cleared, returning the SCI to non-multi-processor reception operation. If the CCR0.RIE bit is set to 1, an SCIn\_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in non-multi-processor Asynchronous mode. The clock used for the multi-processor communication is the same as the clock used in non-multi-processor Asynchronous mode.

## (2) FIFO selected

For data transmission, software must write data to TDR.MPBT (Multi-Processor Bit Transfer) that corresponds to transmit data in TDR.TDAT. For data reception, the multi-processor bit that is part of the receive data is written to RDR.MPB and receive data is written to RDR.RDAT.

When the MPIE bit is set to 1, the following operations are disabled until reception of data in which the multi-processor bit is set to 1:

- Transfer of receive data from the RSR register to the RDR.RDAT register
- Detection of a receive error
- Detection of DR
- Setting of the respective RDRF, ORER, and FER status flags in the CSR register

When the SCI receives a character in which the multi-processor bit is set to 1, the RDR.MPB bit is set to 1 and receive data is written to receive-FIFO(RDR.RDAT). The CCR0.MPIE bit is automatically cleared, returning the SCI to normal reception operation. If the CCR0.RIE bit is set to 1, an SCIn\_RXI interrupt is generated.

When the multi-processor format is specified, the parity bit function is disabled. Apart from this, there is no difference from operation in normal Asynchronous mode with FIFO selected.

### 30.4.1 Multi-Processor Serial Data Transmission

#### (1) Non-FIFO selected

[Figure 30.36](#) shows an example flow of multi-processor data transmission. In the ID transmission cycle, the ID must be transmitted with the TDR.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in Asynchronous mode.

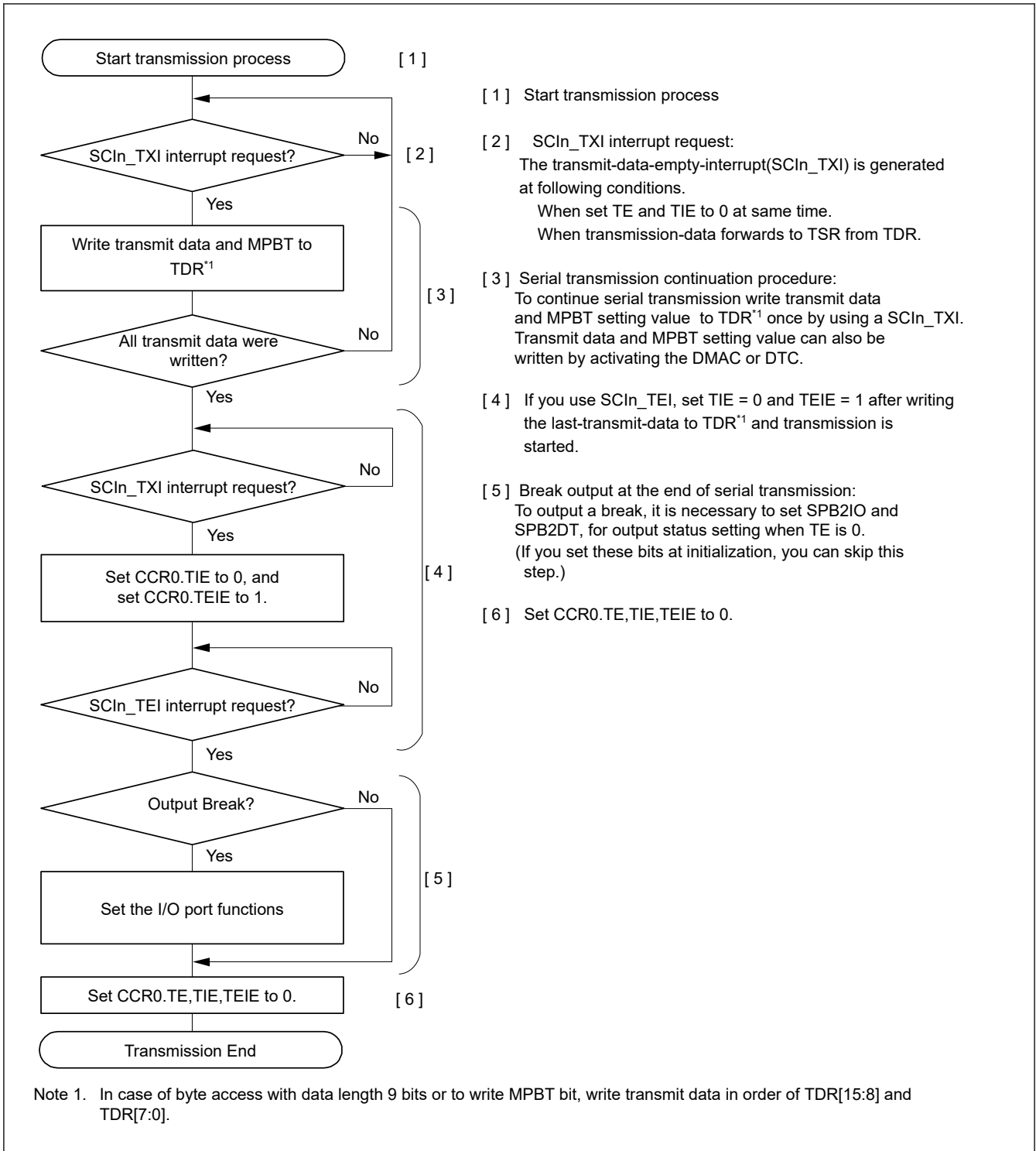


Figure 30.36 Example flow of multi-processor serial transmission with non-FIFO selected

(2) FIFO selected

Figure 30.37 shows an example of data format that is written to transmit-FIFO (TDR) in multi-processor mode. The TDR.MPBT bit is set to 1. Data is set to transmit-FIFO (TDR) with the correct data length. Write 0 for unused bits.

Data Length	Register setting		Transmit data in TDR[15:0]														
	CCR3. CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1
7 bit	1	1	-	-	-	-	-	-	MPBT	-	-	TDAT[6:0]					
8 bit	1	0	-	-	-	-	-	-	MPBT	-	TDAT[7:0]						
9 bit	0	Don't Care	-	-	-	-	-	-	MPBT	TDAT[8:0]							

Note: —: Invalid. The write value should be 0.

**Figure 30.37 Data format written to transmit-FIFO (TDR) in multi-processor mode with FIFO selected**

Figure 30.38 shows an example flow of multi-processor serial transmission with FIFO selected. In the ID transmission cycle, the ID must be transmitted with the TDR.MPBT bit set to 1. In the data transmission cycle, the data must be transmitted with the MPBT bit set to 0. The rest of the operations are the same as operations in Asynchronous mode with FIFO selected.

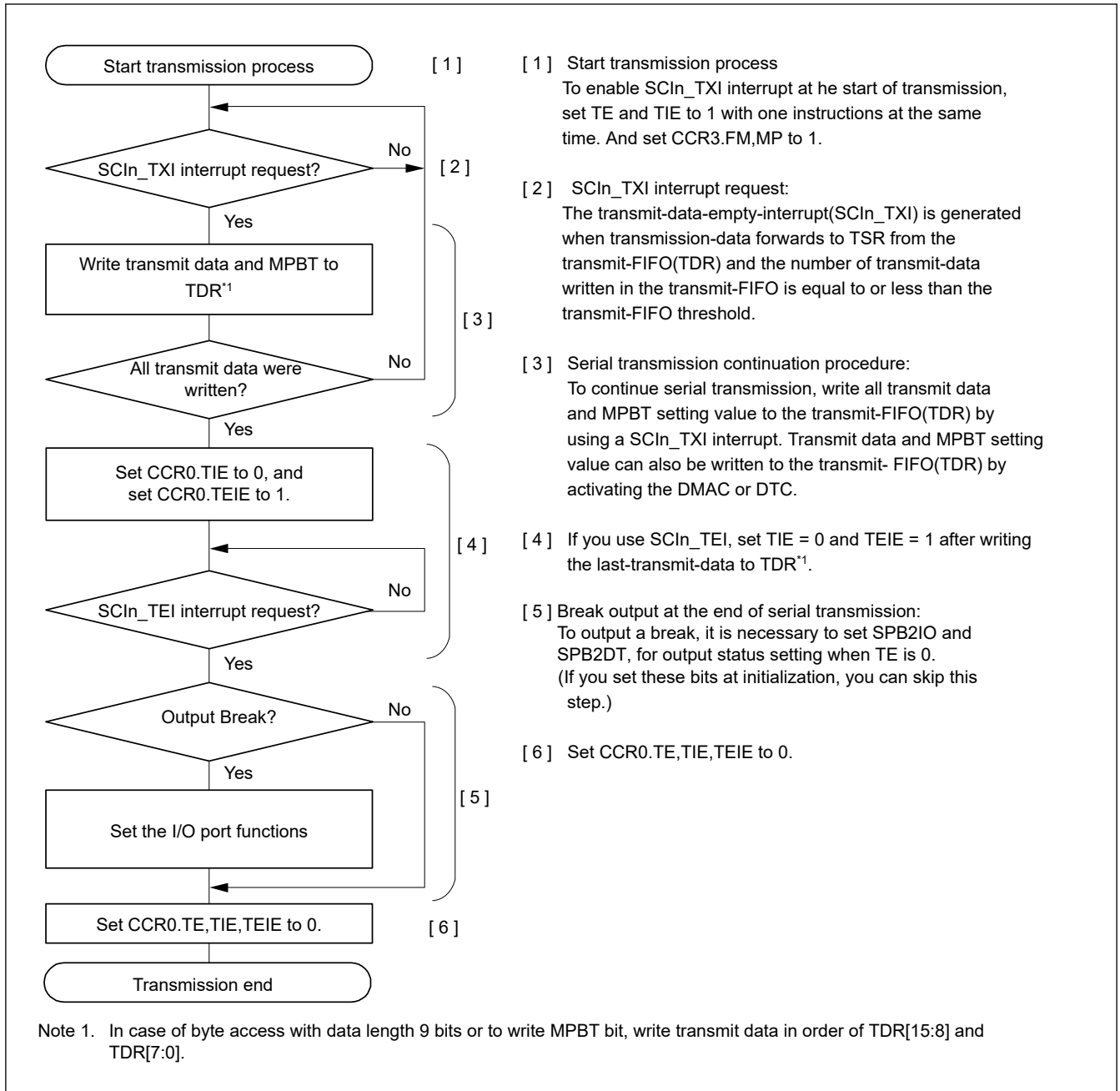


Figure 30.38 Example flow of serial transmission in multi-processor mode with FIFO selected

### 30.4.2 Multi-Processor Serial Data Reception

#### (1) Non-FIFO selected

Figure 30.40 and Figure 30.41 are example flows of multi-processor serial reception. When the CCR0.MPIE bit is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to the RDR register, and the SCIn\_RXI interrupt request is generated. The rest of the operations are the same as operations in Asynchronous mode.

Figure 30.39 shows an example operation for data reception.

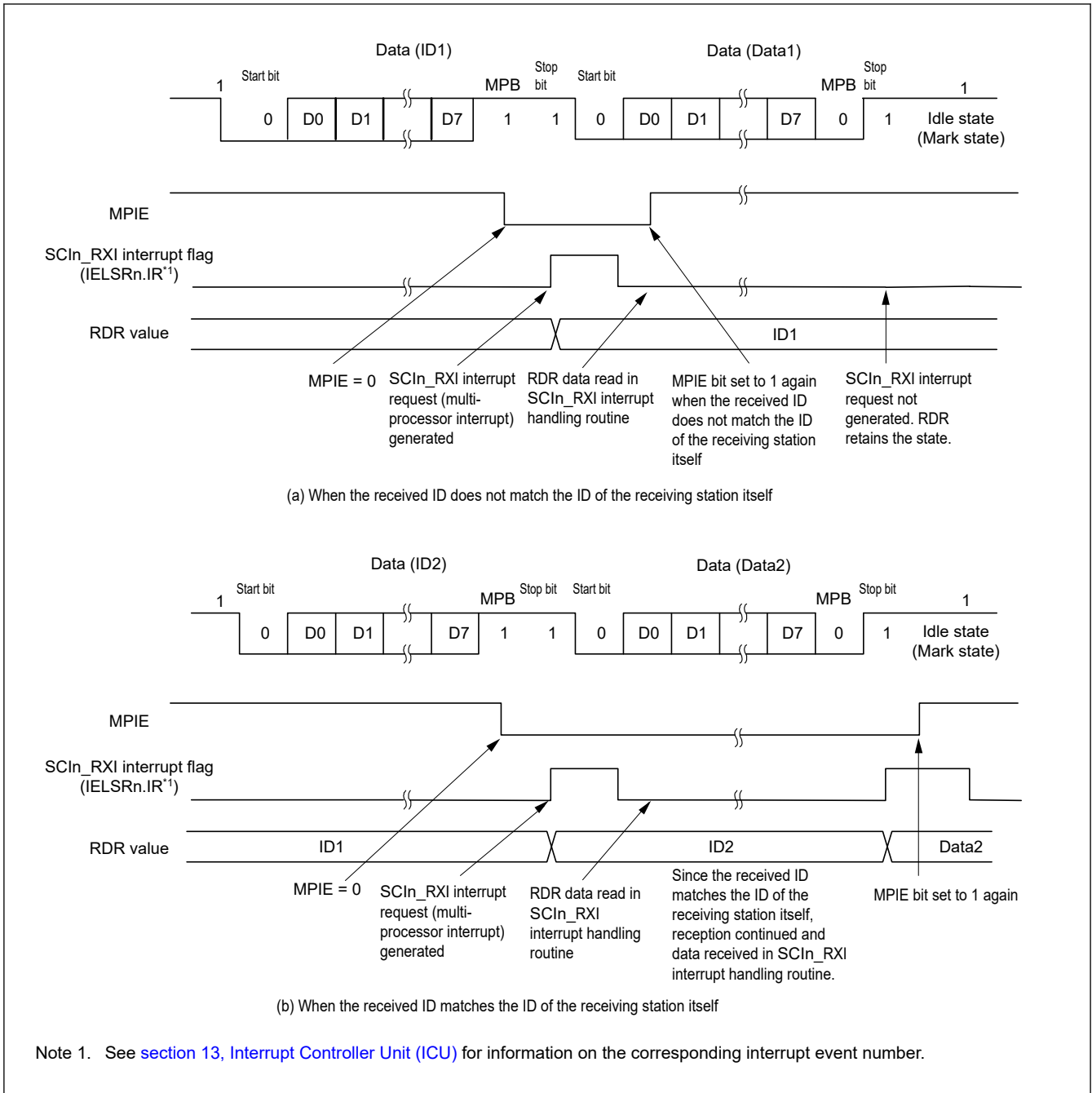


Figure 30.39 Example of SCI reception with 8-bit data, multi-processor bit, and 1 stop bit

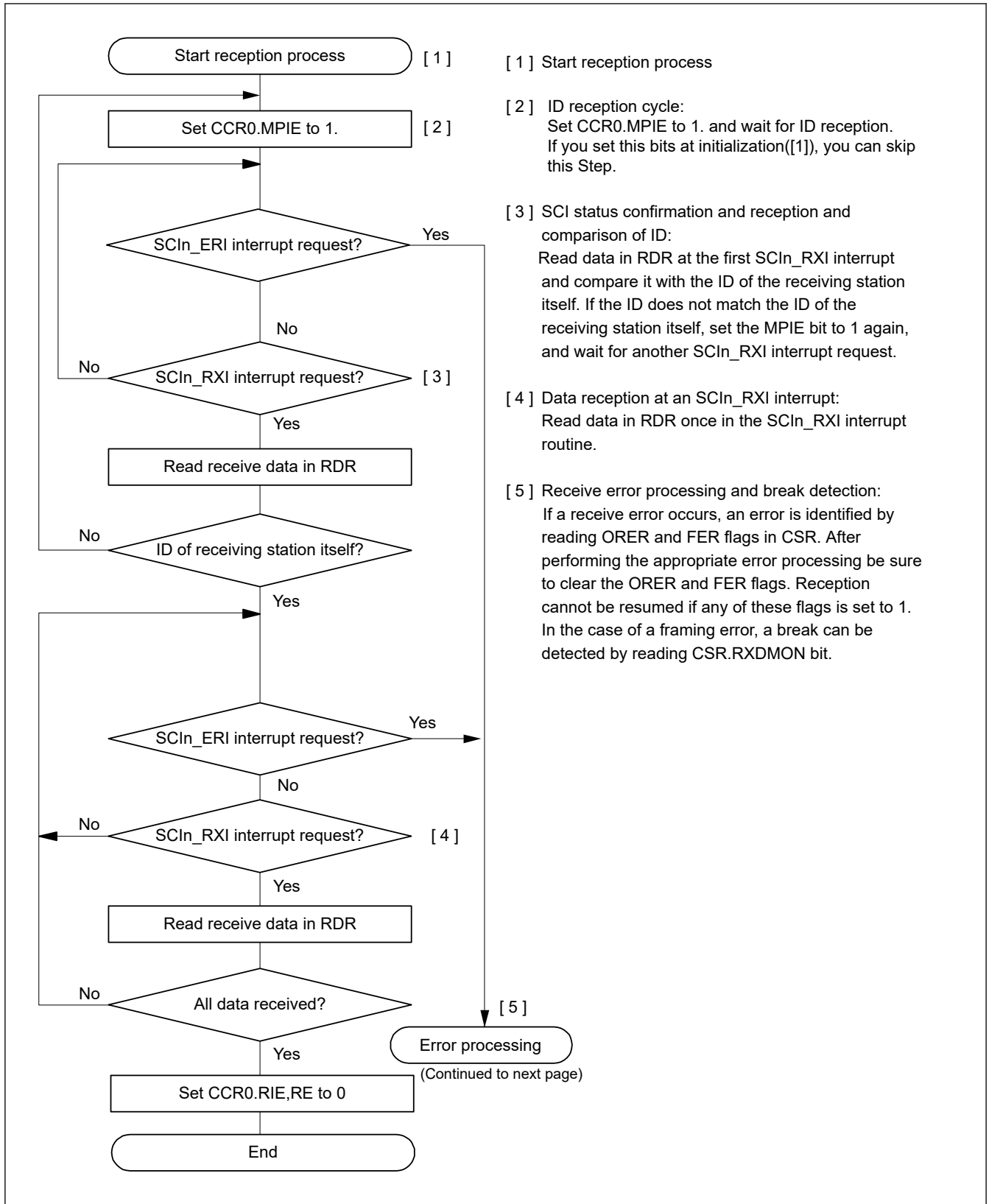


Figure 30.40 Example flow of multi-processor serial reception with non-FIFO selected (1)

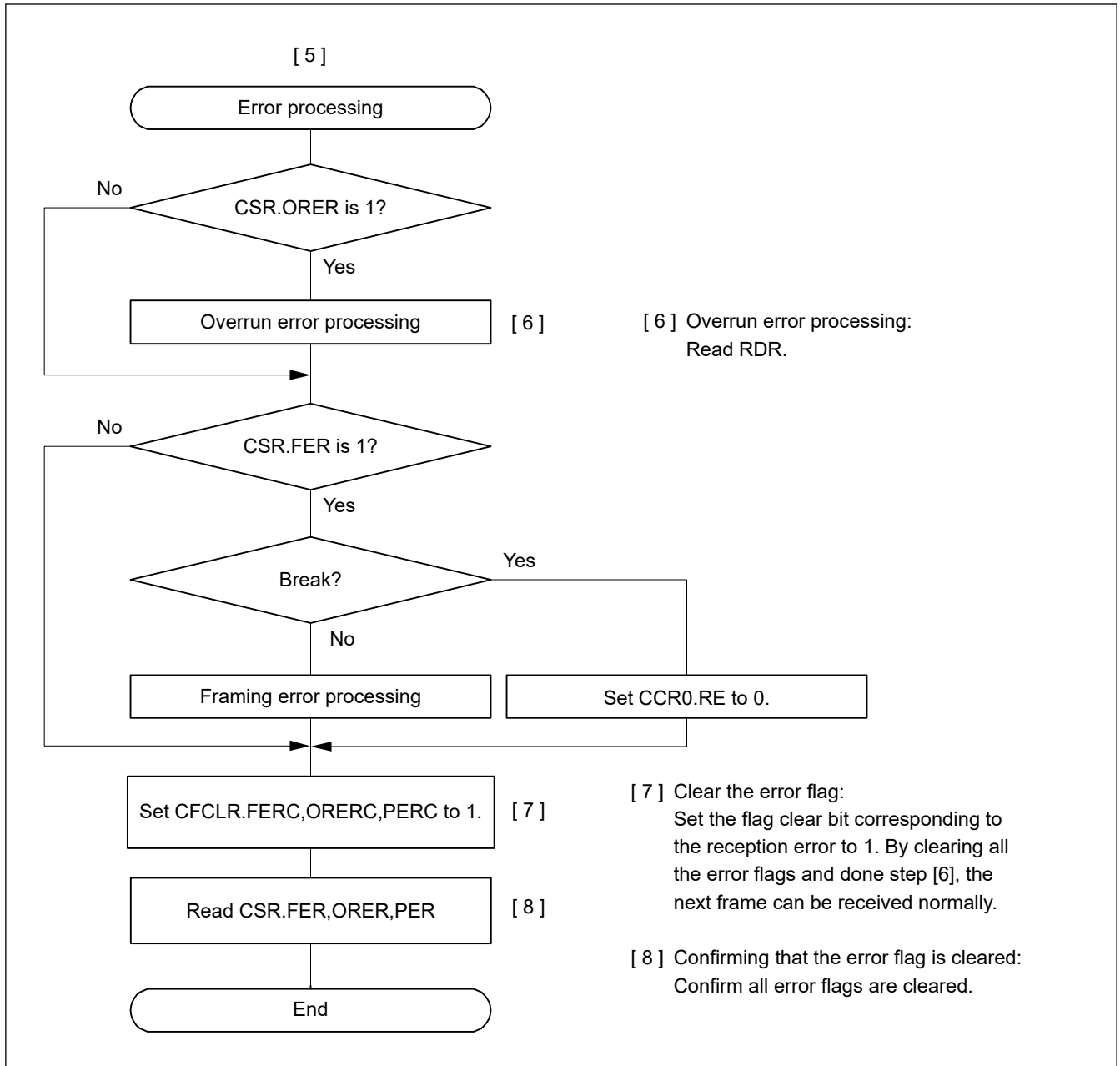


Figure 30.41 Example flow of multi-processor serial reception with non-FIFO selected (2)

(2) FIFO selected

Figure 30.42 shows an example of a data format that is written to receive-FIFO (RDR) in multi-processor mode.

In multi-processor mode, the MPB value that is a part of the receive data is written to the RDR.MPB bit. A value of 0 is written to the RDR.FPER and PER flags. Data is written to receive-FIFO (RDR) with the correct data length. Unused bits are written with 0. When software reads the receive-FIFO (RDR) register, the SCI updates RDR.FFER, FPER, and MPB flags, and receive data (RDAT[8:0]) in receive-FIFO (RDR) with the next data. The FER, PER, and ORER flags in the receive-FIFO (RDR) register always reflect the associated flags in the CSR and FRSR registers.



Data Length	Register Setting		Receive data in RDR[31:0]															
	CCR3.CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7bit	1	1	0	0	0	FFER	FPER	DR	MPB	0	0	RDAT[6:0]						
8bit	1	0	0	0	0	FFER	FPER	DR	MPB	0	RDAT[7:0]							
9bit	0	Don't Care	0	0	0	FFER	FPER	DR	MPB	RDAT[8:0]								
Data Length	CCR3.CHR[1:0]		b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
7bit	1	1	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0
8bit	1	0	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0
9bit	0	Don't Care	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0

Note: When data length is 7bit, it can read always 0 in RDAT[8:7].  
 When data length is 8bit, it can read always 0 in RDAT[8].

**Figure 30.42 Data format stored in receive-FIFO (RDR) in multi-processor mode with FIFO selected**

Figure 30.43 shows an example flow of multi-processor data reception with FIFO selected. When the CCR0.MPIE is set to 1, reading communication data is skipped until reception of communication data in which the multi-processor bit is set to 1. When communication data in which the multi-processor bit is set to 1 is received, the received data, MPB and associated errors are transferred to the receive-FIFO (RDR) register. The CCR0.MPIE bit is automatically cleared and normal reception continues.

If a framing error occurs and the CSR.FER flag is set to 1, the SCI continues data reception. The rest of the operations are the same as operations in Asynchronous mode with FIFO selected.

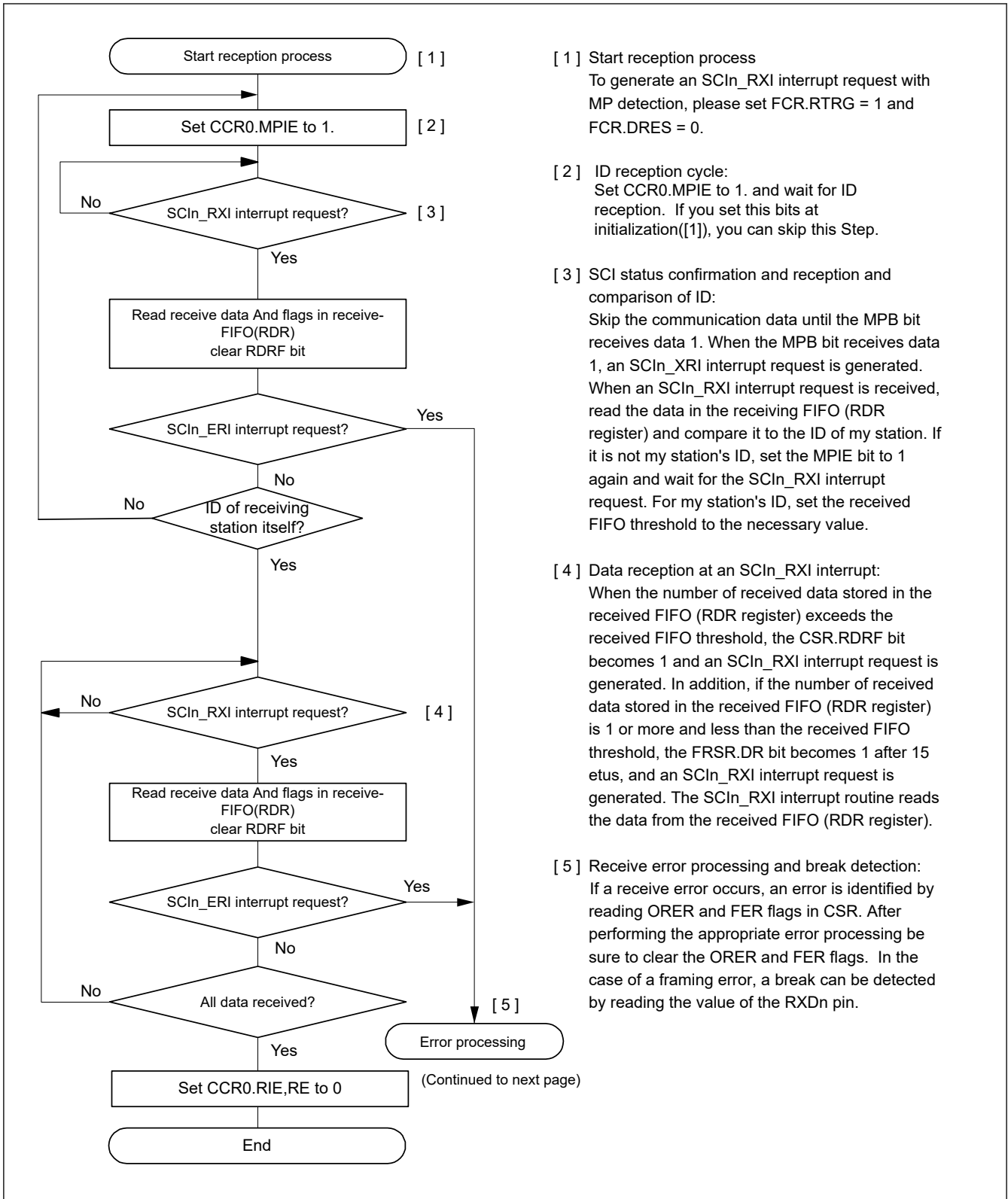


Figure 30.43 Example flow of serial reception in multi-processor mode with FIFO selected

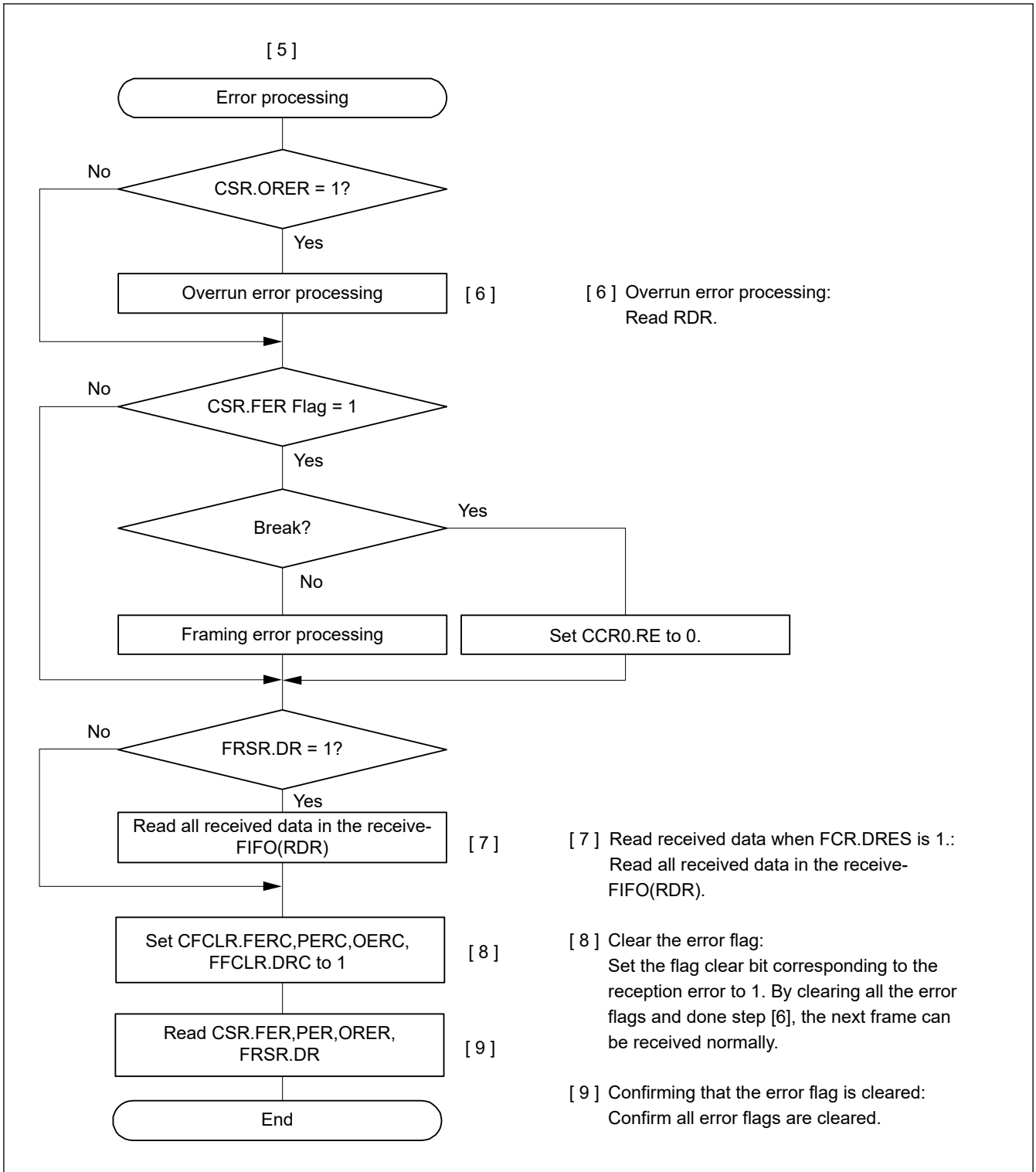
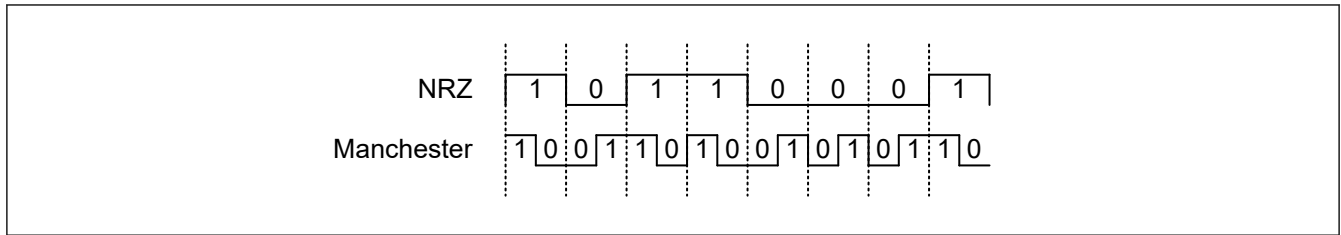


Figure 30.44 Example Flowchart of Serial Reception in Multi-Processor Mode (2) (FIFO selected)

### 30.5 Operation in Manchester mode

In Manchester mode, the transmit or receive serial data is coded in Manchester encoding.

Figure 30.45 shows the conceptual image of Manchester encoding.



**Figure 30.45 Example of Manchester Encoding**

In Manchester mode, a preface and a start bit area are added to the transmit data in the register to configure a transmit frame. For transmission, data is encoded in Manchester encoding. When data is received, frames having the same format as transmitted frames are detected and Manchester decoding is performed.

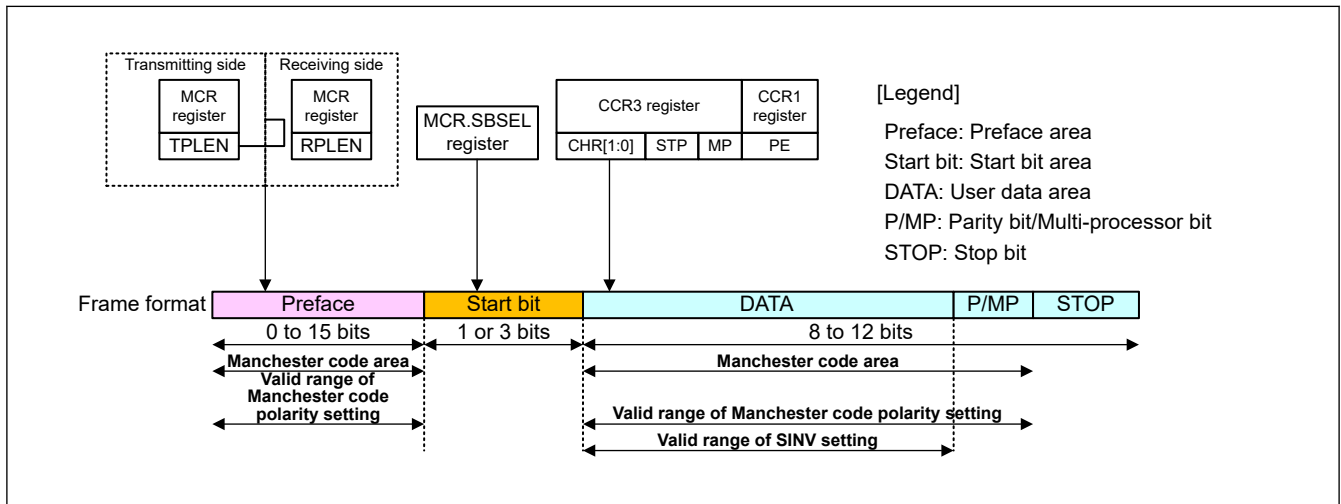
For details on the frame format, see [section 30.5.1. Frame Format](#).

### 30.5.1 Frame Format

[Figure 30.46](#) shows the frame format in Manchester mode.

In the upper half of the figure, relevant setting registers are shown.

The preface area and the data area are encoded in Manchester encoding.



**Figure 30.46 Frame Format in Manchester Mode**

#### (1) Preface area

This is a fixed pattern area located at the beginning of each frame.

Different registers are used to set the preface area for transmission and reception. The preface length is determined by setting MCR.TPLEN[3:0] for transmission. It is determined by setting MCR.RPLEN[3:0] for reception.

If it is set to 0, the transmit preface is disabled and is not added.

If it is set to 1d to 15d, a preface whose length is determined by this setting is added.

(For example, if it is set to 1d, a 1-bit preface is added. If it is set to 15d, a 15-bit preface is added.)

The preface pattern is set with MCR.TPPAT[1:0] for transmission and MCR.RPPAT[1:0] for reception, and is selected from four types of patterns.

[Figure 30.47](#) shows how the preface pattern is set. The preface area and the start bit area are added for each communication frame.

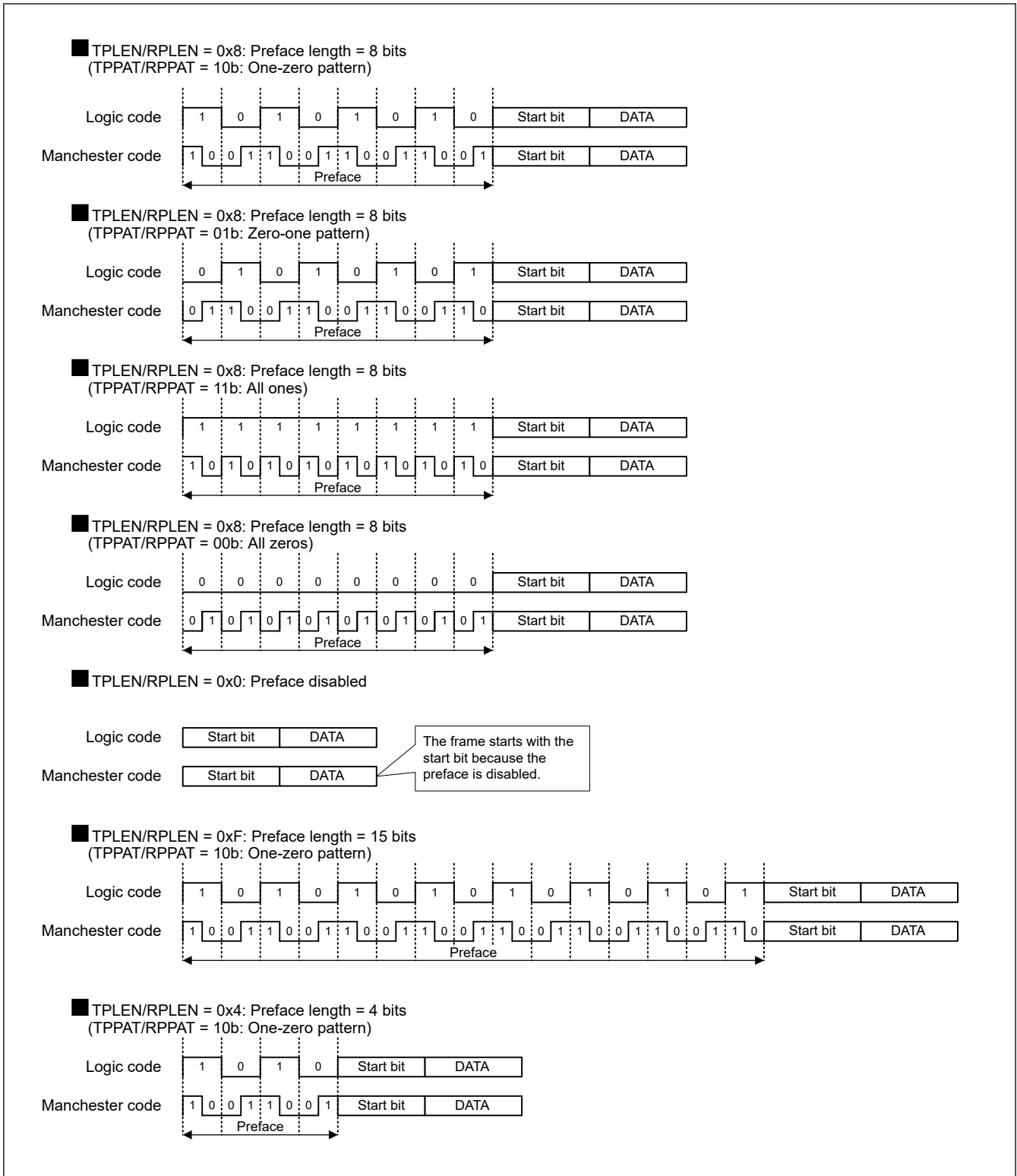


Figure 30.47 Preface Pattern Setting Example

(2) Start bit area

This is an area indicating the start of valid data in a frame. It is added after the preface area.

The start bit length is determined by MCR.SBSEL setting. When MCR.SBSEL = “0”, the start bit length is 1 bit.

When MCR.SBSEL = “1”, the start bit length is 3 bits.

When MCR.SBSEL = “1”, the SYNC type can be selected from command SYNC and data SYNC.

Command SYNC means the three start bits are added as a one-to-zero transition.

Data SYNC means the three start bits are added as a zero-to-one transition.

The SYNC type is determined by the MCR.SYNSEL, MCR.SYNVAL and TDR.TSYNC settings.

(When receiving, the received result is applied to MSR.RSYNC.)

When MCR.SBSEL = "0", the start bit is added as a zero-to-one or one-to-zero transition.

The selection is determined by the MCR.SYNVAL setting.

The MCR.SYNSEL bit specifies the destination to be referred to when setting for transmission.

When the MCR.SYNSEL bit is set to 1, the MCR.SYNVAL setting is referred to. When the MCR.SYNSEL bit is set to 0, the TDR.TSYNC setting is referred to.

Figure 30.48 shows the state of the start bit area according to the settings in the MCR.SYNSEL, MCR.SYNVAL and TDR.TSYNC registers in the case of transmission. Figure 30.49 shows that in the case of reception.

The start bit(s) is not affected by the MCR.TMPOL or MCR.RMPOL setting.

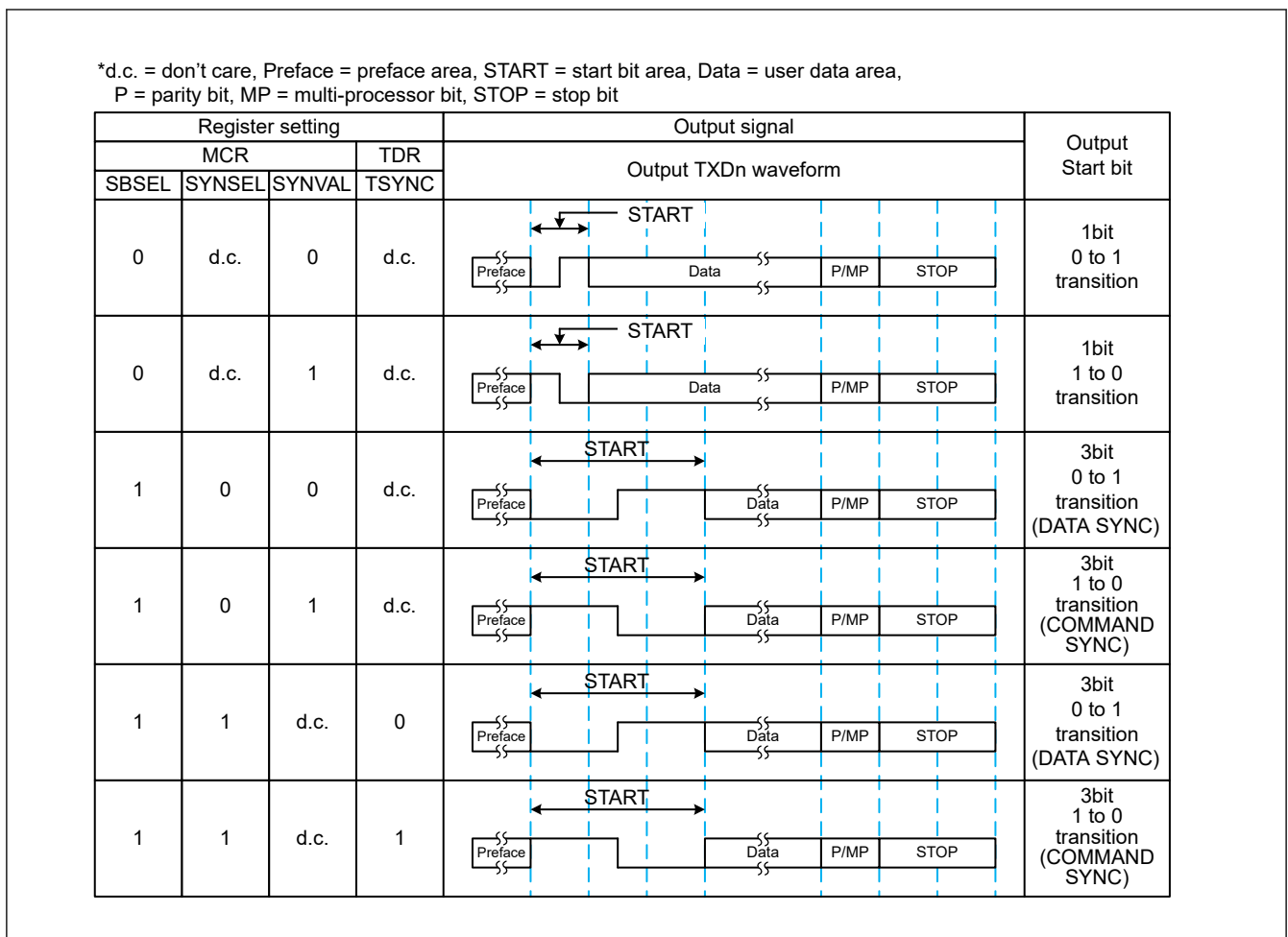


Figure 30.48 Settings Related to and Format of the Start Bit Area at Transmission

d.c. = don't care, Preface = Preface area, START = Start bit area, Data = Data area  
 P = Parity bit, MP = Multi-processor bit, STOP = Stop bit

Register setting				Input signal RXDn input waveform	Start bit detection result <sup>1</sup>	Register indication MSR.RSYNC
MCR		TDR				
SBSEL	SYNSEL	SYNVAL	TSYNC			
0	d.c.	0	d.c.		Normal start bit (1 bit: 0-to-1 transition)	0
					Start bit error	0
					Start bit error	0
					Start bit error	0
0	d.c.	1	d.c.		Start bit error	0
					Normal start bit (1 bit: 1-to-0 transition)	0
					Start bit error	0
					Start bit error	0
1	d.c.	d.c.	d.c.		Start bit error	0
					Start bit error	0
					Data SYNC	0
					Command SYNC	1

Note 1. Data other than the start bit is assumed to be normal.

Figure 30.49 Settings Related to and Judgment of the Start Bit Area at Reception

(3) DATA

Since the format of the data area is the same as that of the Asynchronous mode, see [section 30.3.1. Serial Data Transfer Format](#).

As shown in [Figure 30.45](#), Frame Format in Manchester Mode, the stop bit is not included in the Manchester encoding range.

### 30.5.2 Clock

As the transfer clock in Manchester mode, the clock generated by the on-chip baud rate generator is used by setting the CCR2.CKS[1:0] bit.

Also it is possible to set the oversampling (transfer rate of one-bit period) by CCR2.ABCS bit.

When the CCR2.ABCS bit is set to 0, oversampling x16 is selected with the one-bit period being 16 cycles of the base clock. When the CCR2.ABCS bit is set to 1, oversampling x8 is selected with the one-bit period being 8 cycles of the base clock.

### 30.5.3 Initialization of the SCI in Manchester Mode

Before transferring data, write 0 to CCR0.TE and CCR0.RE (or write the initial value to CCR0 register) and initialize the SCI following the example of flowchart shown in [Figure 30.50](#).

Whenever the operating mode or transfer format is changed, the CCR0 register must be initialized before the change is made.

Note that setting the CCR0.RE bit to 0 initializes none of the ORER, FER, PER, MER, RDRF, SYER, PFER and SBER flags, and the RDR registers.

Note also that switching the value of CCR0.TE from 0 to 1 when CCR0.TIE is 1 generates a SCIn\_TXI interrupt request.



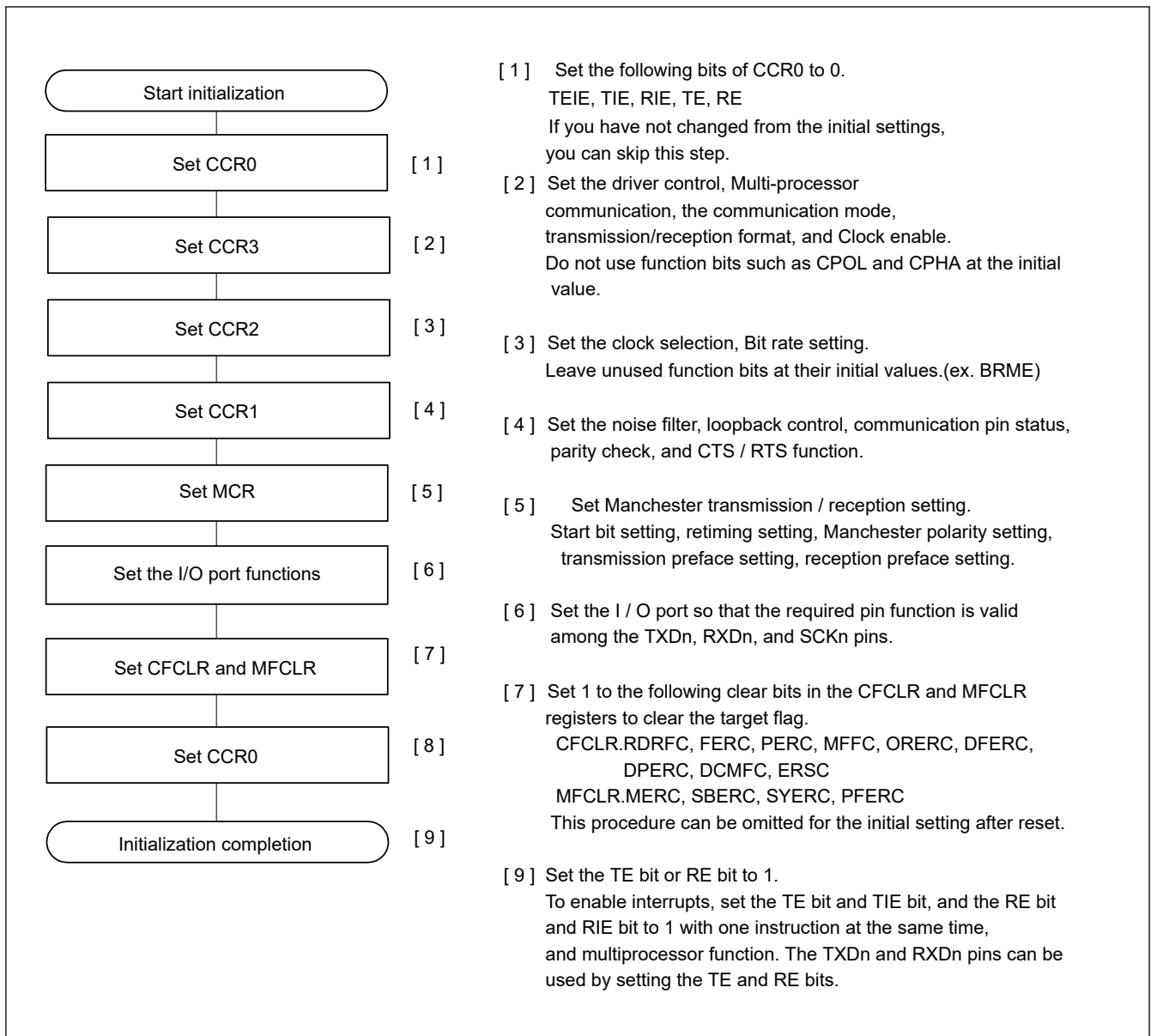


Figure 30.50 SCI Initialization Flow in Manchester Mode

### 30.5.4 Double-speed operation

When the ABCS bit in CCR2 is set to 1 and eight pulses of the base clock for a 1-bit period is selected, the SCI operates on the bit rate twice that of when ABCS is set to 0.

When the BGDM bit in CCR2 is set to 1, the cycle of the base clock is reduced to half and the SCI operates on the bit rate twice that of when ABCS is set to 0.

When the ABCS and the BGDM bits in CCR2 are set to 1, the SCI operates on the bit rate four times that of when the ABCS and the BGDM bits in CCR2 are set to 0.

### 30.5.5 CTS and RTS functions

The CTS function uses input on the CTSn pin in transmission control. Setting the CTSE bit in CCR1 to 1 enables the CTS function. The CTSn\_RTSn pin can be set as a multiplexed pin which allows one pin to be used for either CTS or RTS function, or as dedicated pins with each pin at CTSn pin for CTS function and CTSn\_RTSn pin for RTS function. Use the CTSPEN bit in CCR1 for this setting.

When the CTS function is enabled, reception starts only when the CTSn pin is at the low level.

Applying a high level to the CTSn pin after transmission starts does not affect transmission of the current frame, which continues.

The RTS function uses output on the CTSn\_RTSn pin to request transmission. When the SCI is ready to receive, it outputs a low level to the CTSn\_RTSn pin. Conditions for output of the low level and high level are as follows:

[Conditions for low-level output]

When all conditions listed below are satisfied:

- The value of the RE bit in CCR0 is 1.
- When The SCI is ready to receive next.
  - When there is no received data yet to be read and not receiving
  - All of the following flags are set to 0: CSR.ORER, FER, and PER, MSR.MER, SYER (when SYEREN = 1), PFER (when PFEREN = 1) and SBER flags (when SBEREN = 1).

[Conditions for high-level output]

- When the conditions for low output are not satisfied

### 30.5.6 Serial data transmission in Manchester mode

The SCI encodes data in Manchester encoding and sends the resultant data in Manchester mode.

When the polarity setting (MCR.TMPOL) set to 0, logic 0 is coded as a zero-to-one transition in Manchester code and logic 1 is coded as a one-to-zero transition in Manchester code.

When the polarity setting (MCR.TMPOL) set to 1, logic 0 is coded as a one-to-zero transition in Manchester code and logic 1 is coded as a zero-to-one transition in Manchester code.

For this reason, a level transition occurs with the Manchester encoded data in the middle of individual logic data. (See [Figure 30.45](#)).

The transmitter constructs transmit frames in a specific format by adding a preface area to data and setting the start bit(s) according to the polarity setting and sends resultant serial data.

For details on the frame format, see [section 30.5.1. Frame Format](#).

[Figure 30.51](#) shows the flowchart in transmission. At transmission starts, set the CCR0.TIE and CCR0.TE bits to 1 simultaneously with one instruction. Then, a SCIn\_TXI interrupt request is generated. [Figure 30.52](#), [Figure 30.53](#), and [Figure 30.54](#) show examples of the operation for serial transmission in Manchester mode.

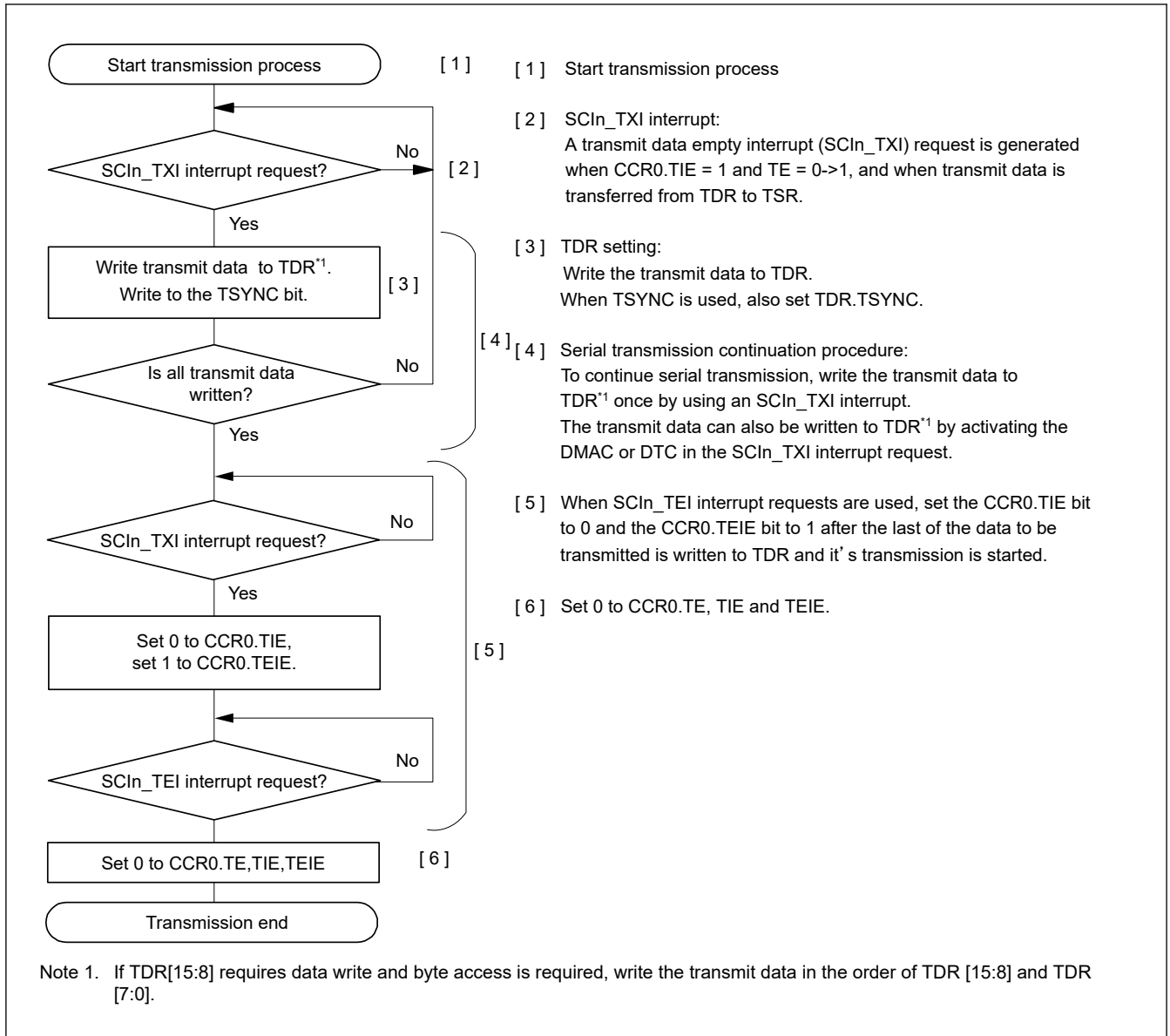
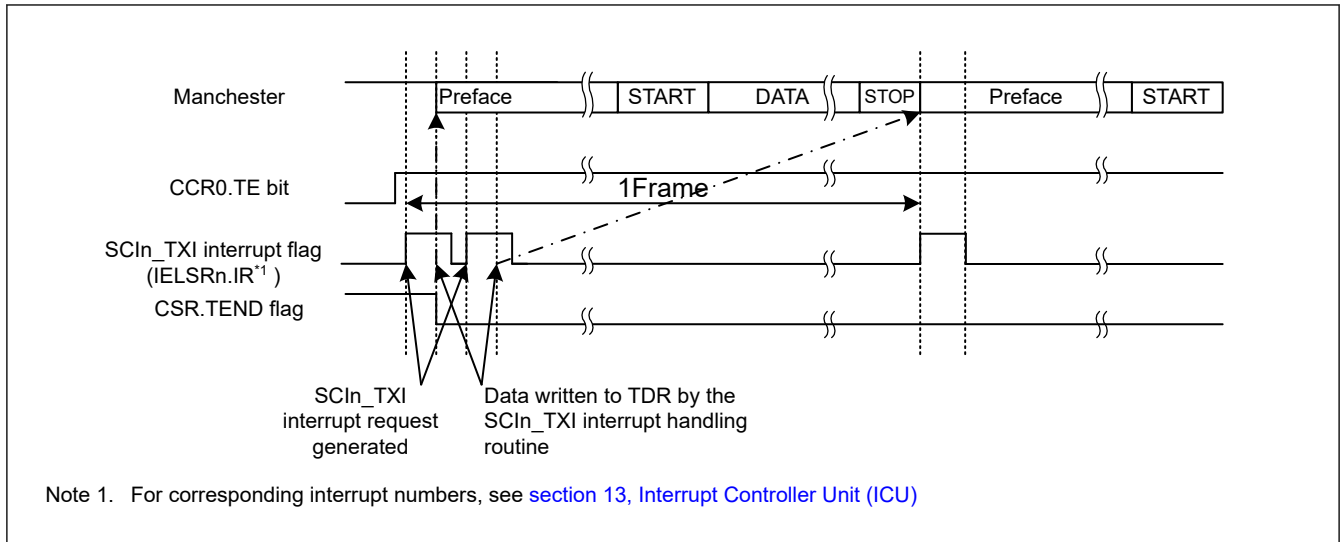
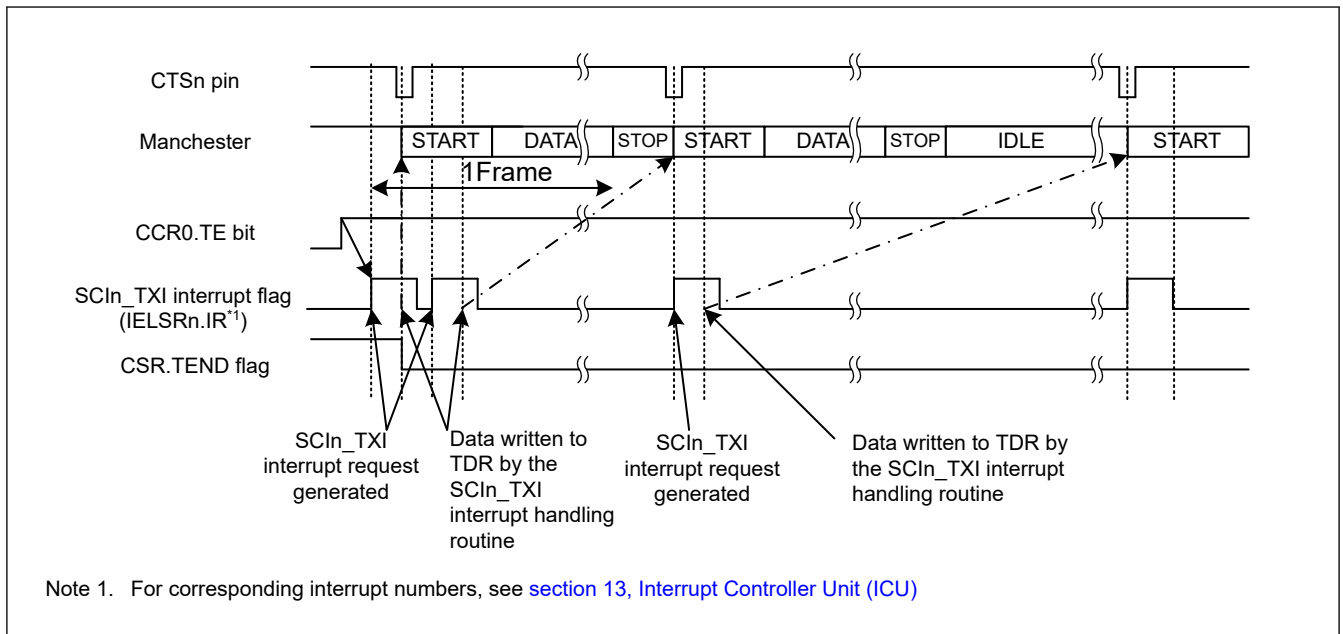


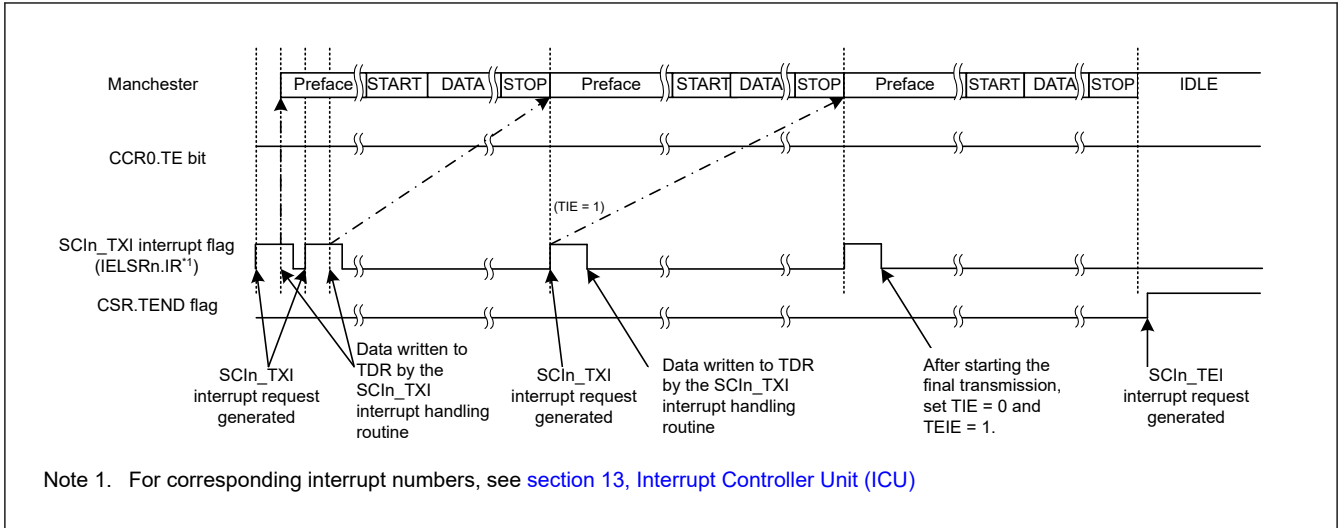
Figure 30.51 Example of Serial Transmission Flowchart in Manchester Mode



**Figure 30.52 Example of Start-of-Transmission Operation for Serial Transmission in Manchester mode (with Preface but Without the CTS Function)**



**Figure 30.53 Example of Start-of-Transmission Operation for Serial Transmission in Manchester Mode (Without Preface but with the CTS Function)**



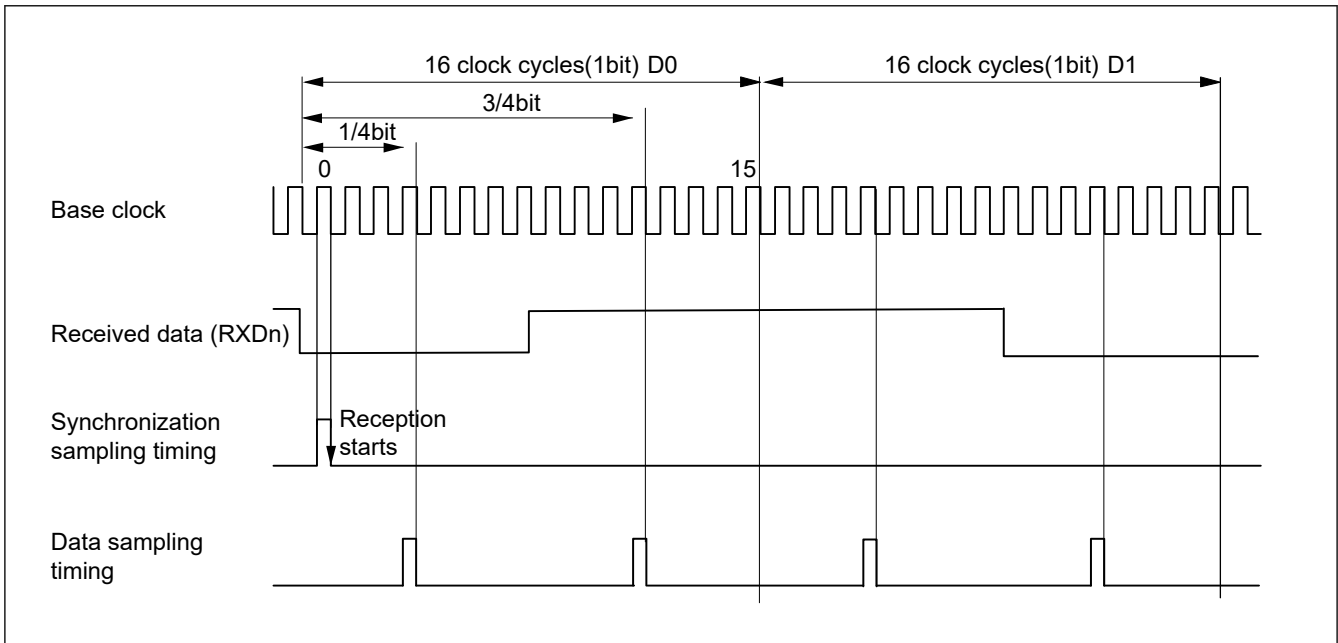
**Figure 30.54 Example of End-of-Transmission Operation for Serial Transmission in Manchester Mode (with Preface but Without the CTS Function)**

### 30.5.7 Serial Data Reception in Manchester Mode

In Manchester mode, the SCI operates on a base clock with a frequency of 16 times<sup>\*1</sup> the bit rate. Reception starts by sampling the falling edges of received data at the base clock. As shown in Figure 30.55, reception starts at a falling edge of the received data and it continues if the received data keeps low for the duration of 1/4 bit. If the received data goes high within the duration of 1/4 bit, the SCI judges it as an error and waits for a falling edge again.

If a high level is expected in the first half of a bit in the received data, the SCI judges a low level that continues for one base clock cycle as an error and ignores the change to the low level.

Note 1. This is the case when CCR2.ABCS = 0. When CCR2.ABCS = 1, the SCI operates on a base clock with a frequency of 8 times the bit rate.



**Figure 30.55 Data Reception Sampling Timing in Manchester Mode**

In Manchester mode, data reception starts with detection of a preface and start bit area.

The SCI checks the input from the RXDn pin to see whether a preface is added based on the value of MCR.RPLEN.

If the preface is disabled (MCR.RPLEN = 0), it moves on to the detection of a start bit area without detecting a preface.

When a preface is enabled, it identifies a preface pattern setting according to the set value in MCR.RPPAT, and compares it with the RXDn input for a pattern match to detect a preface pattern.

Upon detection of a preface pattern match, it judges it as a normal preface and moves on to the detection of a start bit area.

If detecting a preface pattern mismatch or a Manchester code error in the preface area, it judges it as a preface error and asserts a preface error (PFER).

For start bit detection, the SCI selects an expected value based on the register settings (MCR.SBSEL and SYNVAL), compares it with the RXDn input for a pattern match to detect a start bit area. Upon detection of a start bit pattern match, it judges it as a normal start bit area and moves on to the data processing.

Only when a preface and a start bit area are detected normally, it moves on to the next phase of data reception.

Upon detection of a start bit pattern mismatch, it asserts a start bit error flag (SBER).

In data processing, the SCI shifts the data by the expected received data length based on the register settings (CCR3.CHR[1:0]) through the RSR register. If two sampling points in a bit of the received data are identical, the SCI judges this as a Manchester code error.

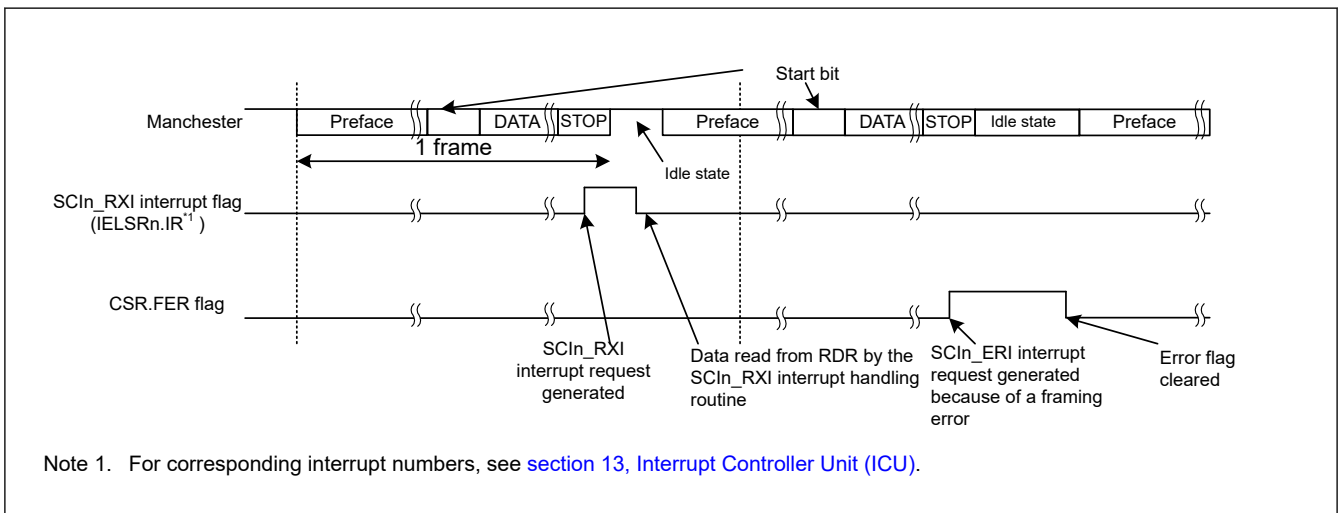
For details, see [section 30.5.11. Errors in Manchester Mode](#) (4).

When the parity function is disabled (CCR1.PE = 0), the SCI moves on to the next phase of stop bit detection. When the parity function is enabled (CCR1.PE = 1), the SCI performs parity checking. If detecting a parity error, it asserts a parity error flag (PER), and then moves on to stop bit detection.

In stop bit detection, the SCI checks the following in the stop bit area of the received frame:

It has two sampling points in a bit. If both points are at the high level, the bit is recognized as a normal stop bit and the data is stored in the RDR register. At least one low-level point is judged as an abnormal stop bit, causing a framing error flag (FER) to be set. Even when an error is detected, the received data is stored in the RDR register as abnormal data.

[Figure 30.56](#) shows an example of the operation for serial data reception in Manchester mode.



**Figure 30.56 Example of Operation for Serial Data Reception in Manchester mode (with a Preface)**

For the state of each status flag in the CCR0 register and RXDn input processing when a receive error is detected, see [section 30.5.11. Errors in Manchester Mode](#).

If a receive error is detected, an SCIn\_ERI interrupt request is generated but an SCIn\_RXI interrupt request is not generated.

Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, PER, MER, SYER\*1, PFER\*1, and SBER\*1 flags to 0 before resuming reception. Also, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the CCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.

[Figure 30.57](#) and [Figure 30.58](#) show examples of serial data reception flowchart in Manchester mode.

Note 1. Effective when the corresponding bit is enabled.

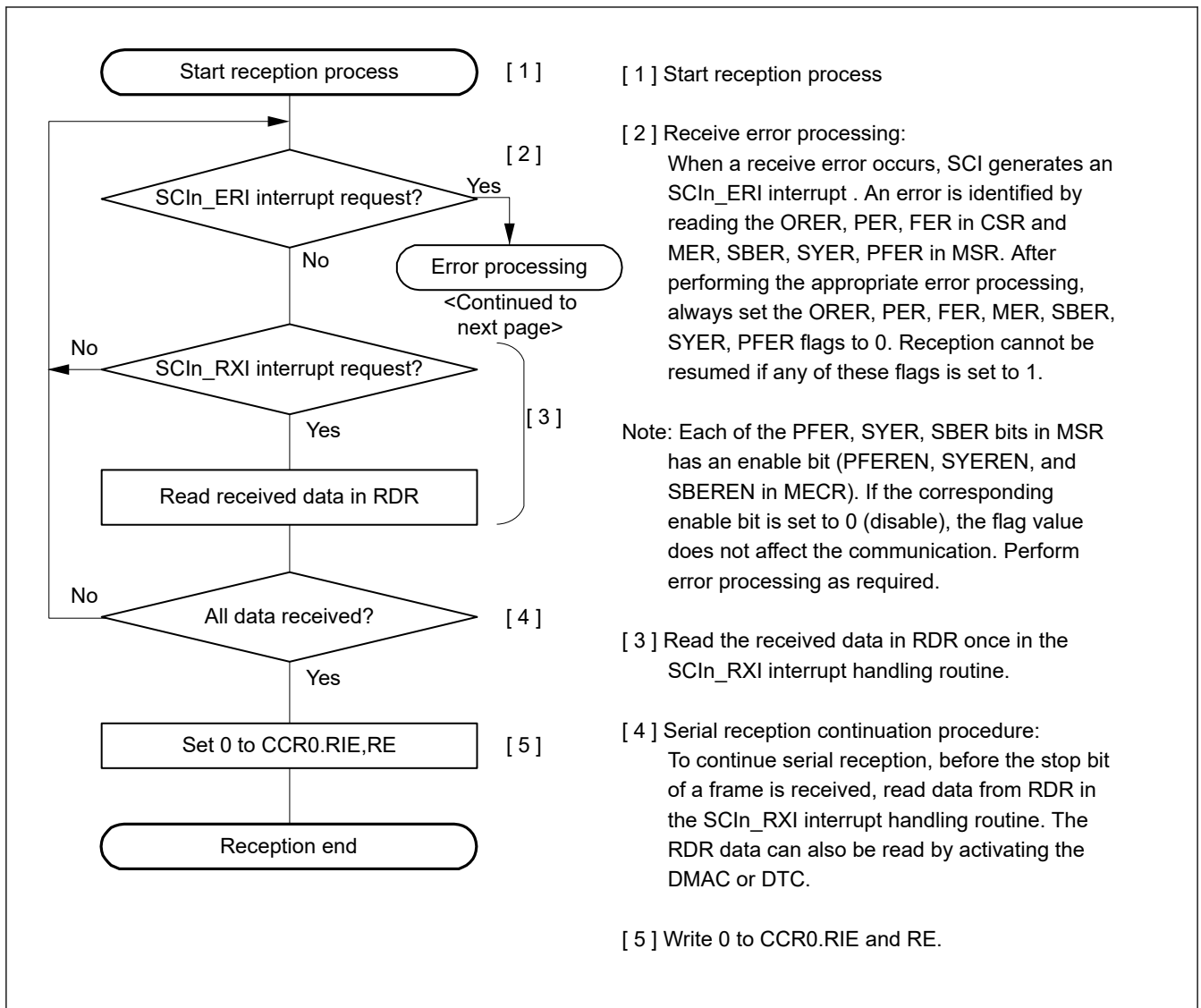


Figure 30.57 Example of Serial Data Reception Flowchart in Manchester Mode (Normal Reception)

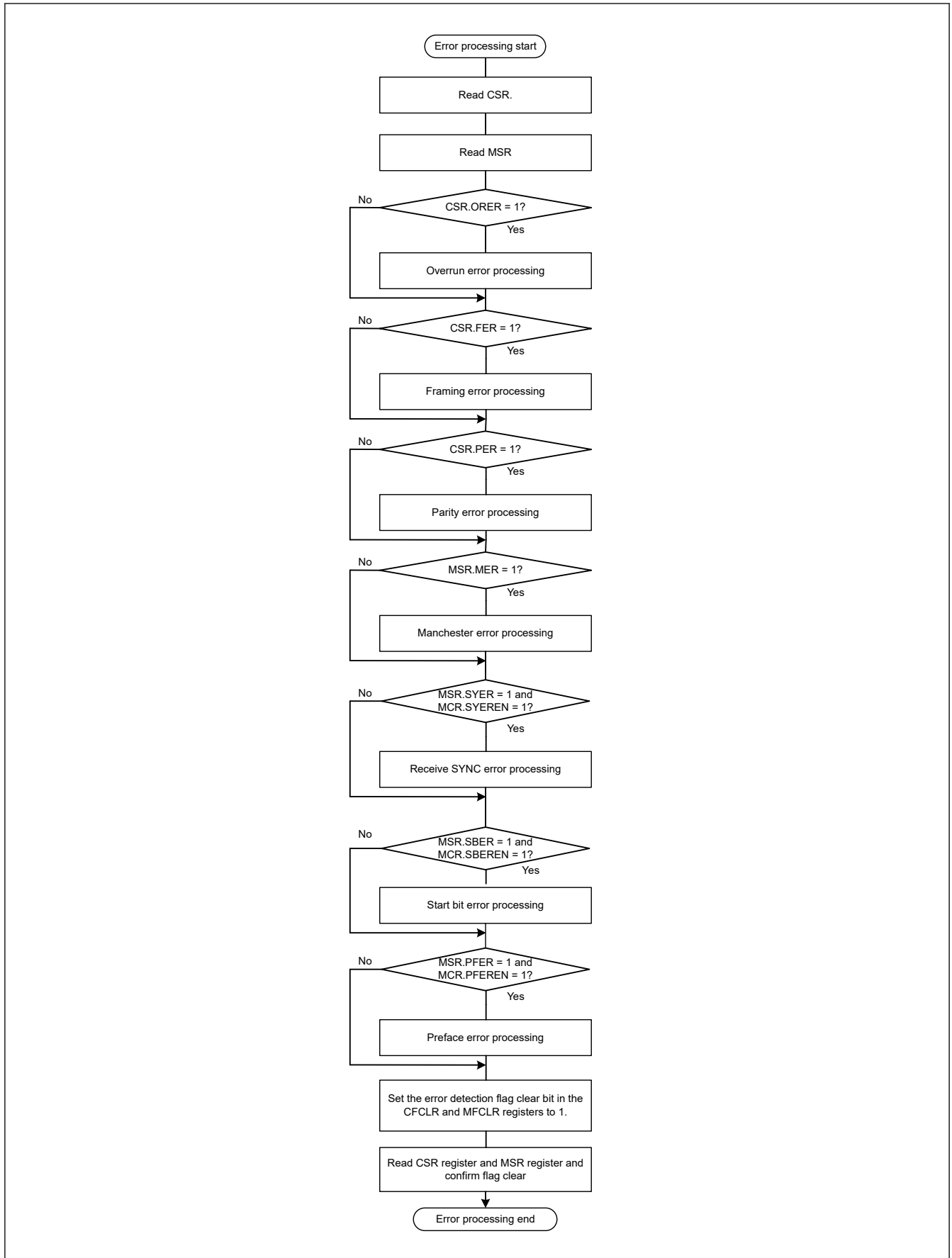


Figure 30.58 Example of Serial Reception Flowchart in Manchester Mode (Error Processing)



### 30.5.8 Operation When Multi-Processor Bit Is Used

See [section 30.4. Multi-Processor Communication Function](#) (1) for the operation in Manchester mode when using multi-processor mode because the operation is the same.

A preface and a start bit area are added to the frame format in Manchester mode. See [Figure 30.58](#) for error processing in Manchester mode for the reception flowchart ([Figure 30.41](#)). See [Table 30.37](#) for the operation status when detecting various errors.

### 30.5.9 Receive Retiming

This function corrects the timing for each central edge of the bit, taking advantage of the fact that each bit has an edge in the center in Manchester code.

The receive retiming function can be turned on or off by setting the ERTEN bit in the MCR register.

When the receive retiming function is turned off ( $MCR.ERTEN = 0$ ), retiming is not performed, causing misalignment between the internal clock and the RXDn input to be accumulated and the receive margin to be reduced.

When the receive retiming function is turned on ( $MCR.ERTEN = 1$ ), retiming is performed for the preface area, the start bit area<sup>\*1</sup>, and the data area (excluding the stop bit).

Note 1. Retiming is not performed for the start bit area if the preface length is 0 and the start bit length is 3.

As an example, the receive retiming when oversampling x16 is selected is shown below.

When detecting an RXDn input edge two to four cycles before the expected receive cycle, the receive processing is shortened by one sampling CLK cycle.

When detecting a RXDn input edge two to three cycles after the expected receive cycle, the receive processing is extended by one sampling CLK cycle.

(Even if the clock is misaligned with the data by more than two cycles, one cycle is corrected for each bit.)

[Figure 30.59](#) shows the conceptual image of receive retiming range.

When detecting an edge in the tolerance area in the figure, data is received as is without making correction.

When detecting an edge in the SyncJump area in the figure, data is corrected for reception.

When detecting an edge in the SyncError area in the figure, data is received as abnormal data with no correction made.

For a Manchester code error (data matches at the 1/4-phase and 3/4-phase sampling points), the SCI reports a code error.

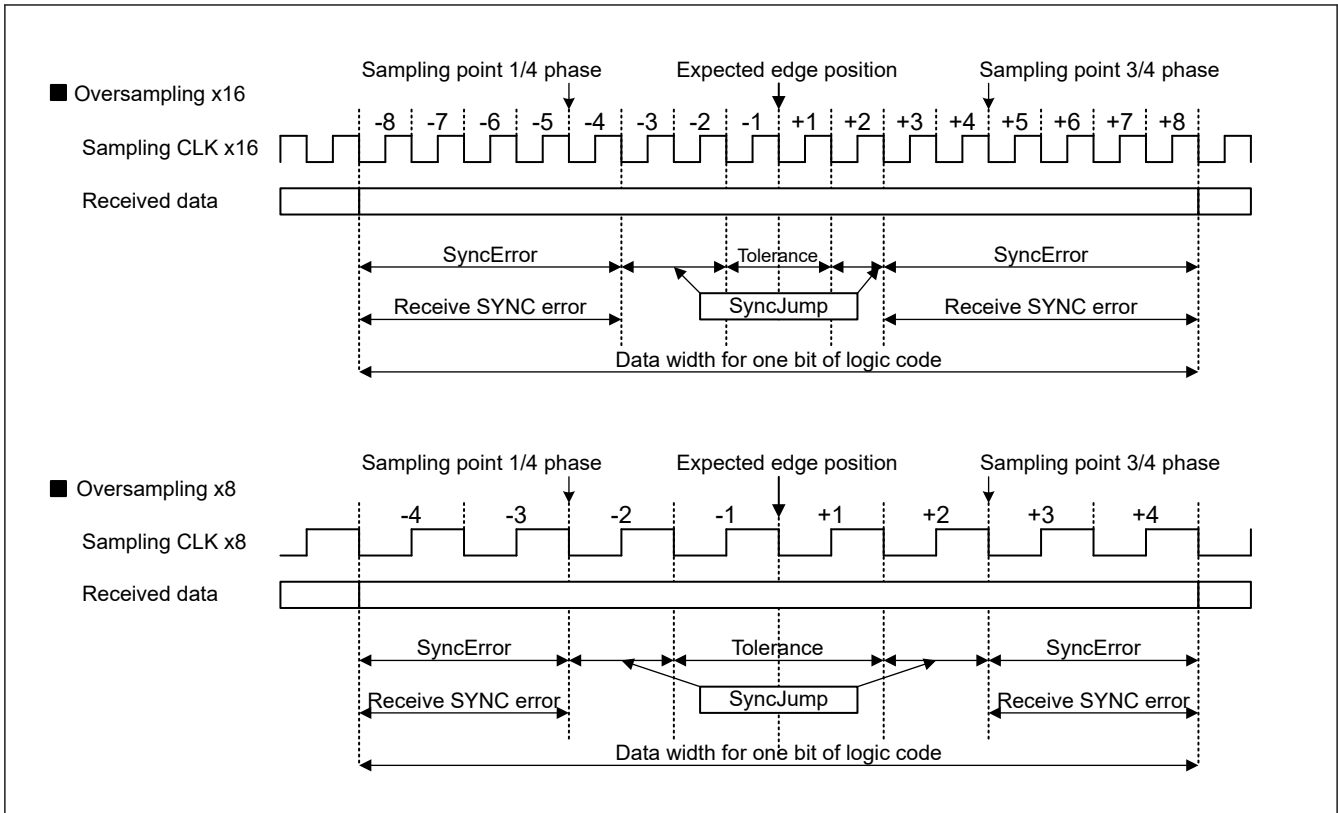


Figure 30.59 Conceptual Image of Reception Retiming Range

### 30.5.10 Polarity Setting for Manchester Code

The polarity of the Manchester code can be set with the Manchester Control Register (MCR).

It can be set separately for transmission and reception. Use the MCR.TMPOL bit to set the polarity for transmission and the MCR.RMPOL bit to set the polarity for reception.

The Manchester code polarity setting is valid for the preface area, the data area, and the parity or multi-processor area.

When the initial settings (TMPOL/RMPOL = 0) are used for the polarity of Manchester code, logic 0 is encoded as a zero-to-one transition in Manchester code and logic 1 is encoded as a one-to-zero transition in Manchester code. If the settings are changed to TMPOL/RMPOL = 1, logic 0 is encoded as a one-to-zero transition in Manchester code and logic 1 is encoded as a zero-to-one transition in Manchester code. Figure 30.60 shows the conceptual image of the settings and operation.

Separately from the function above, the transmitted and received data in the data area can be inverted by the transmitted/received data inversion function (CCR3.SINV). Since the polarity of Manchester code (MCR.TMPOL/RMPOL) can be set separately from the transmitted/received data invert function (CCR3.SINV), if both are set to inversion (MCR.TMPOL/RMPOL = 1 and CCR3.SINV = 1), the transmitted and received data are set to initial state (inversion + inversion = normal).

The polarity of the start bit area can be set by a register different from the ones mentioned above.

Since a different register is used, the polarity of the start bit area is not affected by the polarity setting for Manchester code mentioned above.

For details on the setting for the start bit area, see section 30.5.1. Frame Format (2).

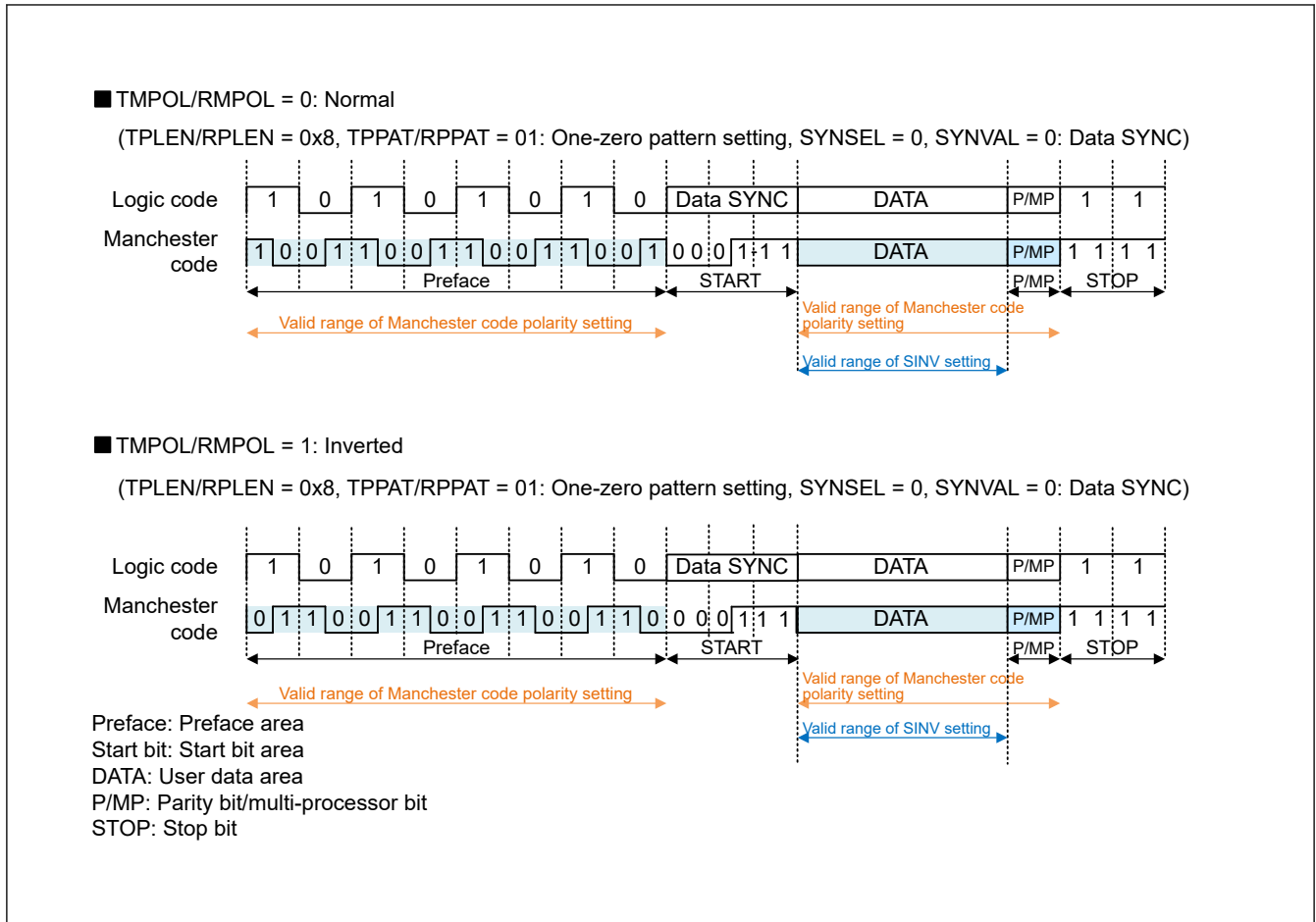


Figure 30.60 Valid Range of the Manchester Code Polarity Setting

### 30.5.11 Errors in Manchester Mode

There are the following errors in Manchester mode:

1. Parity error
2. Over run error
3. Framing error
4. Manchester error
5. Preface error
6. Start Bit error
7. Receive SYNC error

For errors (1) to (3), see [section 30.3.9. Serial Data Reception in Asynchronous Mode \(1\)](#) because they are the same as in Asynchronous mode.

Each errors are judged in each area, but they are reflected on flags and operations at the timing of 3/4-bit sampling of the STOP bit area. If a preface error or start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the previous information.

[Table 30.35](#) lists the states of the serial status register when detecting errors and judgment about whether to store data in the RDR.

[Table 30.36](#) lists the errors that can be detected in each area of a Manchester frame.

If a Preface error or Start bit error is detected, subsequent data will not be received. Therefore, no other error detection is performed, and the error flag holds the result of the previous frame reception. Also, if an error is detected in the previous

frame, data will not be received, but errors in the pre-face area and start bit area will update that flag. [Table 30.37](#) shows the flags and actions in this case.

#### (4) Manchester error

A Manchester error is generated when a Manchester code error is detected.

In Manchester code, there must be an edge (transition) in the center of the bit.

In the data area of a received frame (including the parity/multi-processor bit), the values of the 1/4-bit and 3/4-bit sampling points are checked in each received 1-bit data, and a Manchester code error is determined if these two values match.

If a Manchester code error is detected, the Manchester error flag (MSR.MER) is asserted.

If a Manchester error occurs, it is handled as an interrupt source and event source. If a Manchester error is detected, the next reception is not performed until the corresponding error flag is cleared.

#### (5) Preface error

A preface error is generated when the preface pattern does not match or a Manchester code error is detected in the preface area. If a preface error is detected, the preface error flag (MSR.PFER) is asserted.

It is possible to set whether to use this error flag as an interrupt source with the setting of the MCR register.

When MCR.PFEREN = 1, a preface error is handled as an interrupt source or event source. If a preface error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.PFEREN = 0, a preface error is not handled as an interrupt source or event source, and the next reception is not halted. However, a preface error is notified to MSR.PFER.

#### (6) Start bit error

A start bit error is generated when a mismatch is detected between the start bit area in the received frame and the preset start bit pattern. Upon detection of a start bit error, a start bit error flag (MSR.SBER) is asserted.

It is possible to set whether to use the start bit error as an interrupt source with the setting of the MCR register.

When MCR.SBEREN = 1, a start bit error is handled as an interrupt source or event source. If a start bit error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.SBEREN = 0, a start bit error is not handled as an interrupt source or event source, and the next reception is not halted. However, a start bit error is notified to MSR.SBER.

#### (7) Receive SYNC error

When the receive retiming function described in [section 30.5.9. Receive Retiming](#) is enabled, the receive retiming operation is performed.

If no edges are detected within the receive retiming range (SyncError area in [Figure 30.59](#)) when receive timing operation is being performed, a receive SYNC error is generated. Upon detection of a receive SYNC error, a receive SYNC error flag (MSR.SYER) is asserted. In areas not subject to retiming, receive SYNC errors are not detected.

The preface area<sup>\*1</sup>, the start bit area<sup>\*1,\*2</sup>, and the data area (excluding the stop bit) for which receive retiming operation is performed are checked.

It is possible to set whether to use the receive SYNC error as an interrupt source with the setting of the MCR register.

When MCR.SYEREN = 1, a receive SYNC error is handled as an interrupt source or event source. If a receive SYNC error is detected, the next reception is not performed until the corresponding error flag is cleared.

When MCR.SYEREN = 0, a receive SYNC error is not handled as an interrupt source or event source, and the next reception is not halted. However, a receive SYNC error is notified to MSR.SYER.

Note 1. In the case of a frame that starts with a pattern that expects the first half of the bit to be High, it is excluded from retiming.

Note 2. In the start bit area, when there is no preface length and 3 bit start bit is set, it is not subject to retiming.

Also, the 1st bit and the 2nd bit in the start bit area when 3 bit start bit is set are not subject to retiming.

**Table 30.35 Flags in the CSR Register and Receive Data Handling in Manchester Mode**

Flag in the CSR register			Flag in the MRS register				Received data	Received error status (SCI <sub>n</sub> _ERI interrupt / event generation)
ORER	FER	PER	MER	SBER <sup>*1</sup>	PFER <sup>*1</sup>	SYER		
0	0	0	0	0	0	0	transfer to RDR	No error
0	1	0	0	0	0	0	transfer to RDR	Framing error
0	0	1	0	0	0	0	transfer to RDR	Parity error
0	1	1	0	0	0	0	transfer to RDR	Framing error + Parity error
0	0	0	1	0	0	0	transfer to RDR	Manchester error
0	1	0	1	0	0	0	transfer to RDR	Framing error + Manchester error
0	0	1	1	0	0	0	transfer to RDR	Parity error + Manchester error
0	1	1	1	0	0	0	transfer to RDR	Framing error + Parity error + Manchester error
1	0	0	0	0	0	0	Lost	Overrun error
1	1	0	0	0	0	0	Lost	Overrun error + Framing error
1	0	1	0	0	0	0	Lost	Overrun error + Parity error
1	1	1	0	0	0	0	Lost	Overrun error + Framing error + Parity error
1	0	0	1	0	0	0	Lost	Overrun error + Manchester error
1	1	0	1	0	0	0	Lost	Overrun error + Framing error + Manchester error
1	0	1	1	0	0	0	Lost	Overrun error + Parity error + Manchester error
1	1	1	1	0	0	0	Lost	Overrun error + Framing error + Parity error + Manchester error
0	Combination of above			0	0	1	transfer to RDR	Errors above + Receive SYNC error <sup>*2</sup>
1	Combination of above			0	0	1	Lost	Errors above + Receive SYNC error <sup>*2</sup>
hold	hold	hold	hold	0	1	0	Lost	Preface error <sup>*3</sup>
hold	hold	hold	hold	1	0	0	Lost	Start bit error <sup>*3</sup>
hold	hold	hold	hold	0	1	1	Lost	Preface error <sup>*3</sup> + Receive SYNC error <sup>*2</sup>
hold	hold	hold	hold	1	0	1	Lost	Start bit error <sup>*3</sup> + Receive SYNC error <sup>*2</sup>

Note 1. Start bit error and Preface error never become 1 at the same time.  
 Note 2. When MCR.SYEREN = 1, SCI<sub>n</sub>\_ERI interrupt / event is generated by SYER factor.  
 Note 3. If MCR.PFEREN = 1 or MCR.SBEREN = 1, an SCI<sub>n</sub>\_ERI interrupt / event is generated when the corresponding flag is set.

**Table 30.36 Errors Detectable in Each Area**

	Preface error (PFER)	Start Bit error (SBER)	Manchester error (MER)	Receive SYNC error (SYER)	Parity error (PER)	Framing error (FER)
Preface area	✓	—	— <sup>*1</sup>	✓ <sup>*2</sup>	—	—
Start Bit area	—	✓	—	✓ <sup>*2</sup>	—	—
Data area	—	—	✓	✓	—	—
Parity area	—	—	✓	✓	✓	—
Multi-processor area	—	—	✓	✓	—	—
Stop Bit area	—	—	—	—	—	✓

Note: ✓: Detected, —: Not detected  
 Note 1. When an Manchester code error occurs in the preface area, it is defined as a preface error.  
 Note 2. It may not be subject to Receive SYNC error detection. For details see the text [section 30.5.11. Errors in Manchester Mode \(7\)](#)

**Table 30.37 Operation status due to presence / absence of error in previous frame and operation status list in multiprocessor mode (1 of 2)**

Previous frame	Each area of the Frame					PFEREN	SBEREN	SYEREN	received data	Error flag	Interrupt request	Event signal
	preface	start bit	data	parity	stop							
No Error	PFER	No Error	Don't Care	Don't Care	Don't Care	0	Don't Care	Don't Care	Lost	set PFER*1	not output	not output
	No SYER*1					1					output	output
No Error	SBER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	0	Don't Care	Lost	set SBER*1	not output	not output
	No SYER*1						1				output	output
SYER No PFER	No Error	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0	transfer to RDR	set SYER	not output	not output
								1	Lost		output	output
No Error	SYER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0	transfer to RDR	set SYER	not output	not output
	No SBER							1	Lost		output	output
No Error	No Error	No Error	SYER		No Error	Don't Care	Don't Care	0	transfer to RDR	set SYER	not output	not output
								1			output	output
No Error	No Error	No Error	MER		No Error	Don't Care	Don't Care	Don't Care	transfer to RDR	set MER	output	output
No Error	No Error	No Error	Don't Care	PER	No Error	Don't Care	Don't Care	Don't Care	transfer to RDR	set PER	output	output
No Error	No Error	No Error	Don't Care	Don't Care	FER	Don't Care	Don't Care	Don't Care	transfer to RDR	set FER	output	output
There is some error ORER						Don't Care	Don't Care	Don't Care	Lost	set some flags*2	output	output
No Error	No Error	No Error	No Error	No Error	No Error ORER	Don't Care	Don't Care	Don't Care	Lost	set ORER	output	output

**Table 30.37 Operation status due to presence / absence of error in previous frame and operation status list in multiprocessor mode (2 of 2)**

Previous frame	Each area of the Frame					PFERE N	SBERE N	SYERE N	received data	Error flag	Interrupt request	Event signal	
	preface	start bit	data	parity	stop								
some error <sup>*3 *6</sup>	PFER	No Error	Don't Care	Don't Care	Don't Care	0	Don't Care	Don't Care	Lost	set PFER <sup>*1</sup>	output <sup>*4</sup>	not output <sup>*5</sup>	
	No SYER <sup>*1</sup>					1							
	No Error	SBER No SYER <sup>*1</sup>	Don't Care	Don't Care	Don't Care	Don't Care	0	Don't Care					set SBER <sup>*1</sup>
							1						
	SYER No PFER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0					set SYER
								1					
	No Error	SYER No SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0					set SYER
								1					
	No Error	No Error	SYER		No Error	Don't Care	Don't Care	0					don't set any flags
								1					
No Error	No Error	MER		No Error	Don't Care	Don't Care	Don't Care						
No Error	No Error	Don't Care	PER	No Error	Don't Care	Don't Care	Don't Care						
No Error	No Error	Don't Care	Don't Care	FER	Don't Care	Don't Care	Don't Care						
There is some error ORER						Don't Care	Don't Care	Don't Care					
No Error	No Error	No Error	No Error	No Error	ORER	Don't Care	Don't Care	Don't Care					

- Note 1. If SYER is detected, the SYER flag is also set. Other operations are as shown in this table.
- Note 2. Other detected error flags including ORER are also set.
- Note 3. If all the error flags are cleared before the STOP bit is judged, the operation will be the same as the case where there is no error in the previous frame of this table.
- Note 4. Since the SCIn\_ERI interrupt request is level output, it remains active due to errors in the previous frame regardless of the presence or absence of error in the relevant frame.
- Note 5. Since the error cause is continuously detected, the SCIn\_ERI event is not newly output regardless of the presence or absence of errors in the relevant frame.
- Note 6. For PFER, SBER, and SYER, when each enable bit is set to disable, it is treated as no error.

**Table 30.38 Operation when MPIE = 1 in multi-processor mode (MPIE = 0)**

MPB <sup>*1</sup>	Each area of the frame					PFERE N	SBERE N	SYERE N	received data	Error flag	Interrupt request	Event signal
	preface	start bit	data	parity	stop							
1	No Error	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	transfer to RDR	set some flags	output <sup>*2</sup>	output <sup>*2</sup>
	No PFER	No SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	0				
	SYER <sup>*3</sup>	SYER <sup>*3</sup>						1	Lost	don't set any flags	not output	not output
	PFER	No Error	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care				
	No Error	SBER	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care	Don't Care				

- Note 1. If the received MPB bit is 0, it is not received the frame, and the operation is the same as lost of the reception data of this table.
- Note 2. If no error is detected, SCIn\_RXI interrupt request or event is output, and if it is detected, SCIn\_ERI interrupt request or event is output.

Note 3. When SYER is detected in the preface area or the start bit area, the behavior of handling as an error depending on the SYEREN bit changes.

## 30.6 Operation in Clock Synchronous Mode

Figure 30.61 shows the data format for clock synchronous serial data communications.

In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. For single-character data transfer, data consists of 8-bit. In clock synchronous mode, no parity bit can be added.

In data transmission when  $CPHA = 1$  and  $CPOL = 1$ , the SCI outputs data from one falling edge of the synchronization clock to the next falling edge. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the transmission line holds the last bit as output state. When the  $CPHA$  bit is 0 in slave mode, the transmission line holds the first bit output state.

Within the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by using a shared communication clock of the transmitter and the receiver. Furthermore, because both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

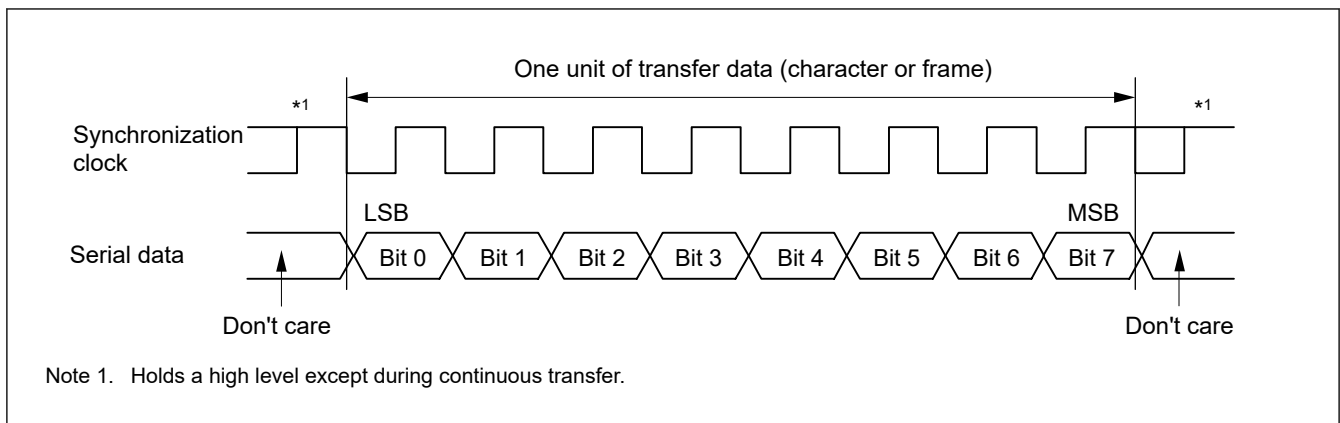


Figure 30.61 Data format in clock synchronous serial communications with LSB-first order

### 30.6.1 Clock

If the maximum speed of  $SCK = 1/2TCLK$  is set in Clock Synchronous and Simple SPI mode, Do not make  $PCLK$  less than half the speed of  $TCLK$ . If  $PCLK$  is made slower than this, malfunction may occur.

#### 1. When the internal clock is selected

When the SCI operates on an internal clock ( $CCR3.CKE[1:0]$  bits are set to 00b or 01b (master mode)), the synchronization clock is output from the  $SCKn$  pin. Eight synchronization clock pulses are output during single-character transmission/reception. When no data transfer is performed, the clock is held at high level.\*<sup>1</sup> In transmission-only or transmission/reception, the synchronization clock is not output unless transmit data is prepared.

When the internal clock is selected, the clock with a delay from the  $SCKn$  signal is used for the master reception sampling clock. This ensures the data setup time and hold time for high-speed communication.

Note 1. The signal is held high while ( $CCR3.CPHA = 0$  and  $CCR3.CPOL = 1$ ) or ( $CCR3.CPHA = 1$  and  $CCR3.CPOL = 1$ ). It is held low while ( $CCR3.CPHA = 0$  and  $CCR3.CPOL = 0$ ) or ( $CCR3.CPHA = 1$  and  $CCR3.CPOL = 0$ ).

#### 2. When the external clock is selected

When the  $CCR3.CKE[1:0]$  bits are set to 10b or 11b (slave mode), data is transmitted and received using the external clock that is input from the  $SCKn$  pin.

### 30.6.2 CTS and RTS Functions

In the CTS function, the  $CTS_n\_RTS_n$  pin input controls the start of data reception or transmission when the clock source is the internal clock. Setting the  $CCR1.CTSE$  bit to 1 enables the CTS function. In clock synchronous communication, the CTS function can be used for the internal clock and the RTS function can be used for the external clock, so the CTS function and RTS function cannot be used at the same time. When the CTS function is enabled, setting the  $CTS_n\_RTS_n$  pin low causes data reception or transmission to start.



Setting the CTSn\_RTsn pin high while the data transmission or reception is in progress does not affect transmission or reception of the current frame.

In the RTS function, the CTSn\_RTsn pin output is used to request the start of data reception or transmission when the clock source is an external synchronizing clock. The CTSn\_RTsn output goes low when serial communication is enabled. Conditions for output of the CTSn\_RTsn low and high are shown as follows:

[Conditions for low output]

Satisfaction of all the following conditions:

#### Non-FIFO selected when all of the following conditions are satisfied

- The value of the CCR0.RE bit or the CCR0.TE bit is 1
- Next serial communication is enabled.
  - No receive data is present before reading and not receiving. (when CCR0.RE bit = 1)
  - When the transmission data written in TDR is ready for transmission.\*1 (when CCR0.TE bit = 1)
- The CSR.ORER flag is 0

Note 1. The CTSn\_RTsn pin will be High after starting transmission.

#### FIFO selected when all of the following conditions are satisfied

- The value of the CCR0.RE bit or the CCR0.TE bit is 1
- Next serial communication is enabled.
  - The number of receive data stored in the receive FIFO (RDR register) is less than the value set in FCR.RSTRG[4:0] (when CCR0.RE bit = 1)
  - When the transmission data written in the transmission FIFO (TDR register) is ready for transmission.\*1 (when CCR0.TE bit = 1)
- The CSR.ORER flag is 0

Note 1. The CTSn\_RTsn pin will be High after the last data transmission starts.

[Condition for high output]

- The conditions for low output are not satisfied

### 30.6.3 SCI Initialization in Clock Synchronous Mode

Before transmitting and receiving data, start by writing the initial value 0x00 to the CCR0 register, then continue through the SCI initialization procedure given in the sections describing non-FIFO and FIFO selection in [section 30.6.2. CTS and RTS Functions](#). Anytime the operating mode or transfer format is to be changed, the CCR0.TE and CCR0.RE must write to 0 before the change can be made.

Note: Setting the CCR0.RE bit to 0 initializes neither the ORER, FER, PER, and RDRF flags in CSR nor the RDR register. When the TE bit is set to 0, the TEND flag for the selected FIFO buffer is not initialized.

Note: Switching the value of the CCR0.TE bit from 1 to 0 when the CCR0.TIE bit is 1 generates an SCIn\_TXI interrupt request.

**Table 30.39 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected (1 of 2)**

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set the CCR0.TEIE, TIE, RIE, TE, and RE bits to 0*1. If you have not changed from the initial settings, you can skip this step.
3	Set FCR	Set the TFRST and RFRST to 1 to empty FIFO. Set the TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits.

**Table 30.39 Example flow of SCI initialization in clock synchronous mode with non-FIFO selected (2 of 2)**

No.	Step Name	Description
4	Set CCR3 except MOD[2:0]	Set CCR3 except of communication mode. <ul style="list-style-type: none"> <li>• FIFO use/no-use</li> <li>• Transmission/reception format</li> <li>• Clock setting</li> <li>• Leave unused bits (CHR[1:0],STP,RXDSEL,MP,DE, ACS0,GM,BLK) at their initial values.</li> </ul>
5	Set CCR3.MOD[2:0]	Set communication mode (MOD[2:0] = 010b) <sup>*2</sup> .
6	Set CCR2	Select clock, set bit rate <sup>*3</sup> . Leave unused bits (BCP[2:0],ABCS, ABCSE, BRME,MDDR[7:0]) at their initial values.
7	Set CCR1	Set up the loop-back function, communication pin status and the CTS/RTS function.
8	Set CCR4	Set up the adjust sampling timing function. Leave unused bits (CMPD[8:0]) at their initial values.
9	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
10	Set CFCLR, FFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERFC, PERFC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC FFCLR.BRKC, DRC
11	Set CCR0	Set the TE or RE bit to 1. <sup>*1 *4</sup> To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time. Setting the TE and RE bits allows TXDn and RXDn to be used.
12	Initialization completion	—

Note 1. In simultaneous transmit and receive operations, the TE and RE bits in CCR0 should both be to 0 or set to 1 simultaneously.

Note 2. Set CPOL and CPHA before setting the communication mode.

Note 3. If you use an external clock, you do not need to set it.

Note 4. When using the internal clock (master), the setting of reception only is prohibited.

### 30.6.4 Serial Data Transmission in Clock Synchronous Mode

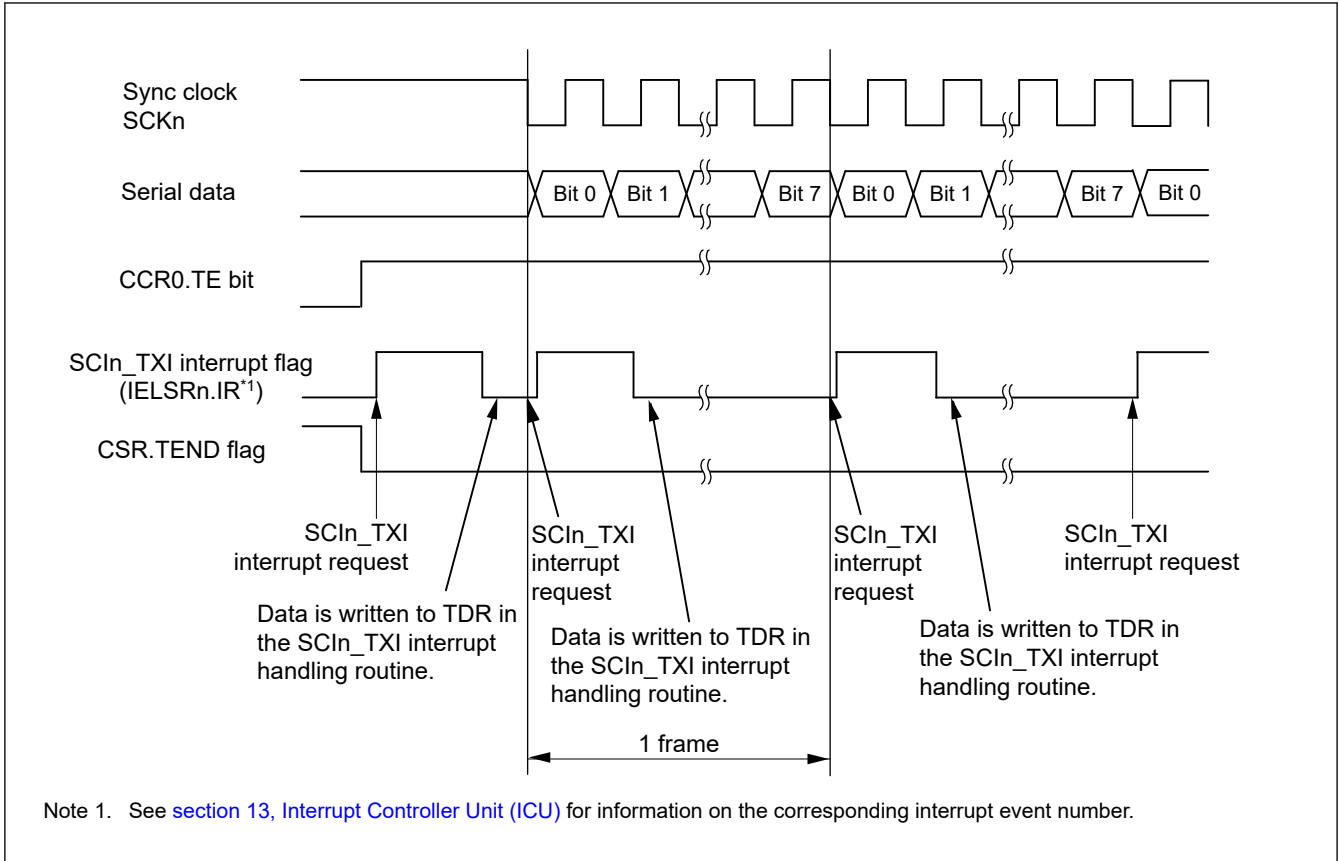
#### (1) Non-FIFO selected

Figure 30.62, Figure 30.63, and Figure 30.64 show examples of serial transmission in clock synchronous mode.

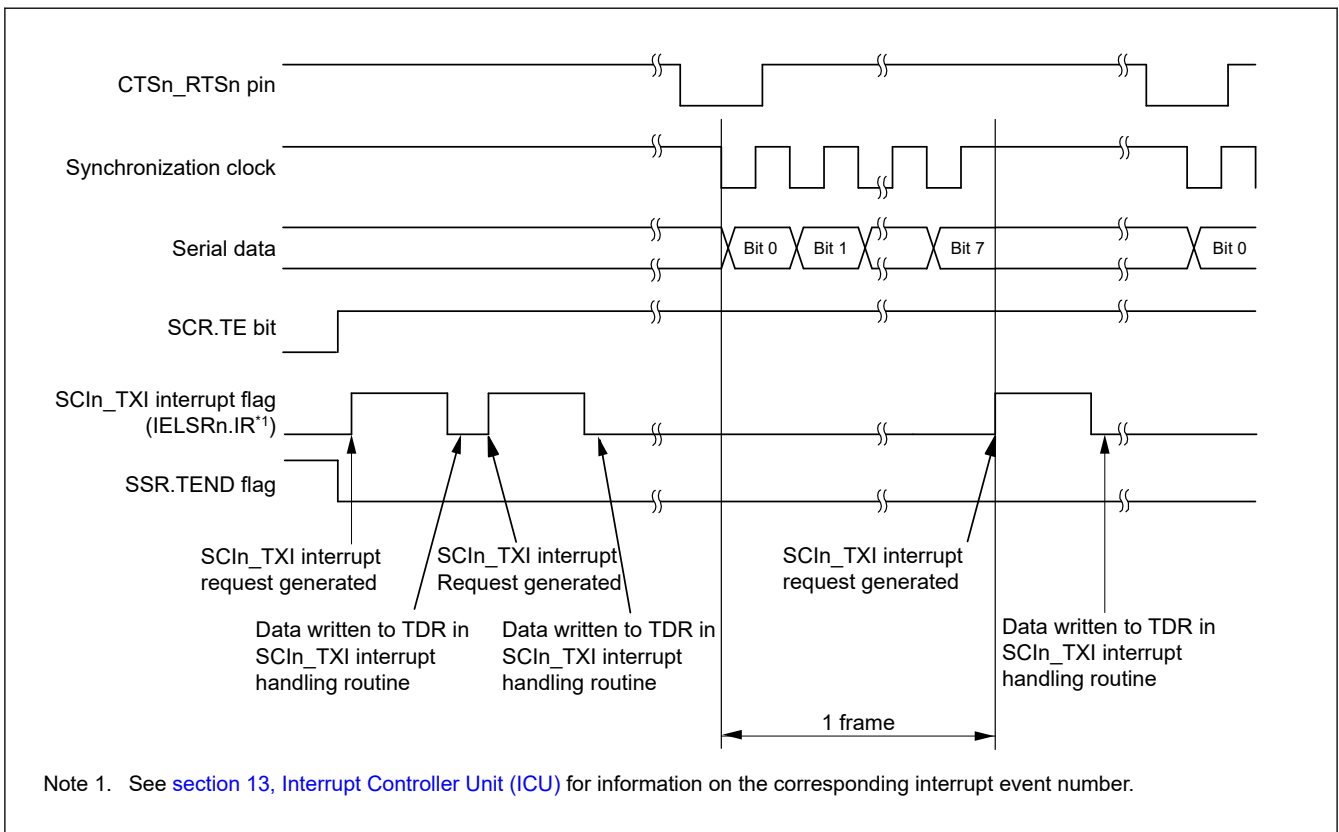
In serial data transmission, the SCI operates as follows:

1. The SCI transfers data from the TDR register to the TSR register when data is written to TDR in the SCIn\_TXI interrupt handling routine. When starting data transmission, set the CCR0.TIE bit and the CCR0.TE bit to 1 simultaneously by a single instruction. Then a SCIn\_TXI interrupt request is generated.
2. After transferring data from TDR to TSR, the SCI starts transmission. When the CCR0.TIE bit is set to 1, an SCIn\_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in the SCIn\_TXI interrupt handling routine before transmission of the current transmit data finishes. When SCIn\_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 and the CCR0.TEIE bit to 1 after the last of the data to be transmitted is written to the TDR register from the handling routine for SCIn\_TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the CCR1.CTSE bit is 1 (CTS function enabled).
4. The SCI checks for update to the TDR register on output of the last bit.
5. When the TDR register is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame starts.
6. If TDR is not updated, the CSR.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the CCR0.TEIE bit is 1, an SCIn\_TEI interrupt request is generated and the SCKn pin is held high.

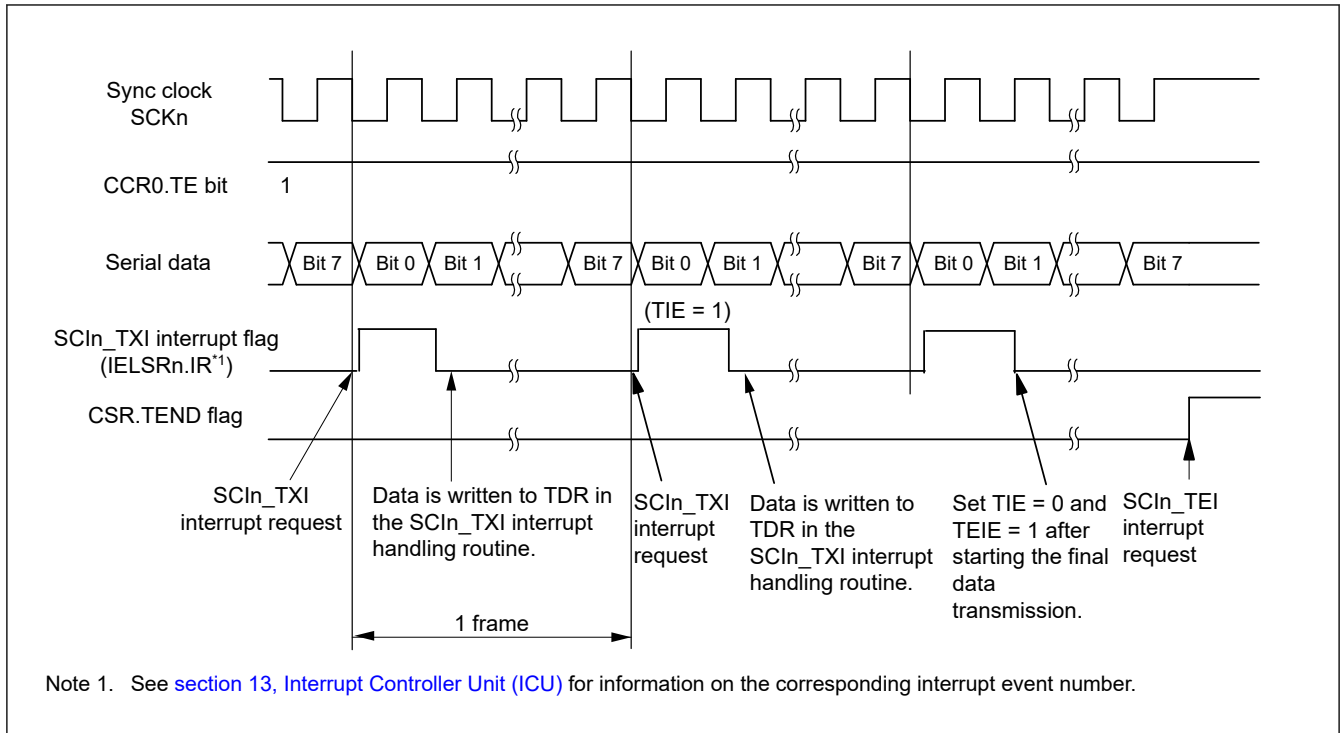
Figure 30.62, Figure 30.63, and Figure 30.64 show examples of serial data transmission.



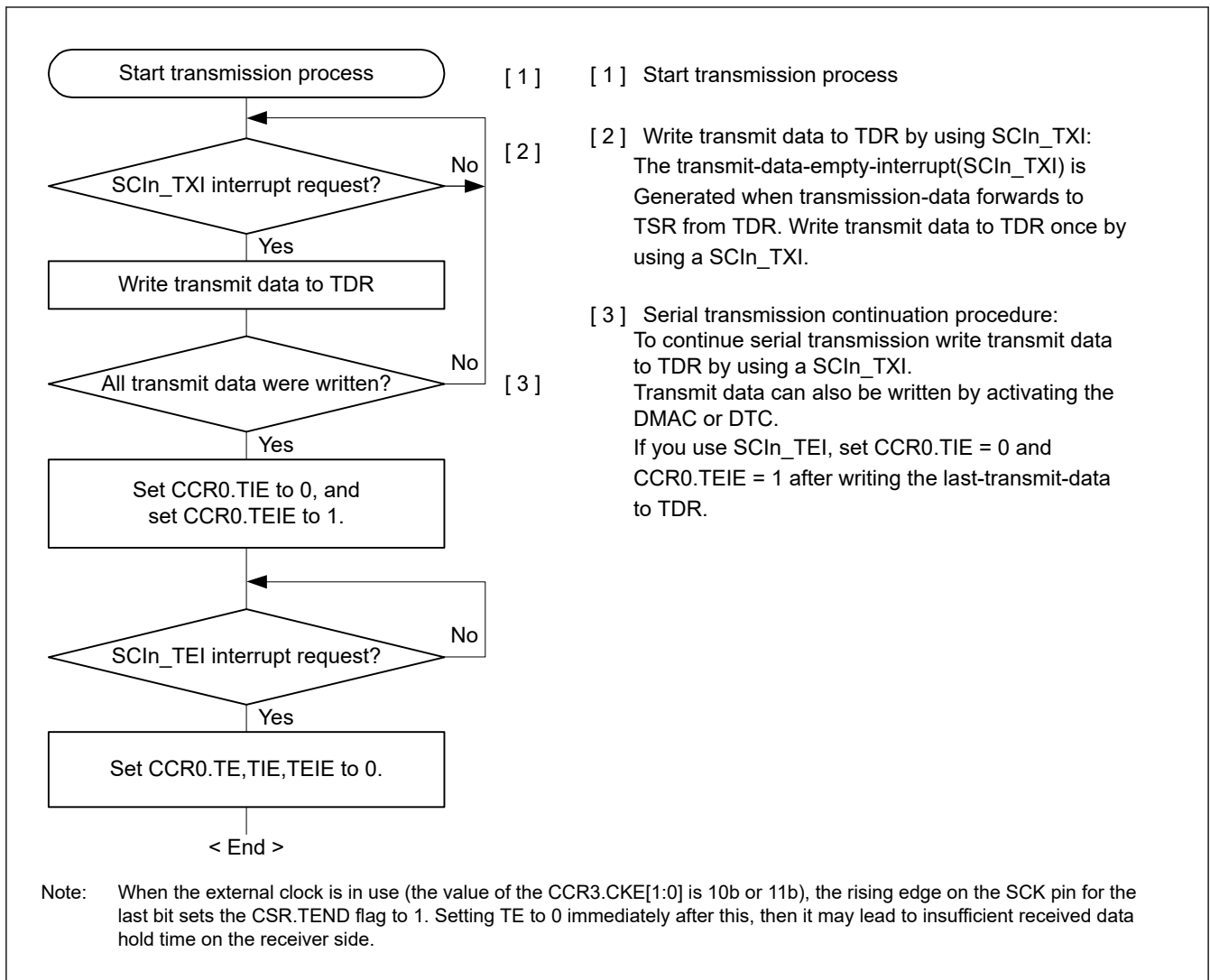
**Figure 30.62** Example of serial data transmission in clock synchronous mode when the CTS function is not used at the beginning of transmission



**Figure 30.63** Example of serial data transmission in clock synchronous mode when the CTS function is used at the beginning of transmission



**Figure 30.64** Example of serial data transmission in clock synchronous mode from the middle of transmission until transmission completion



**Figure 30.65 Example flow of serial transmission in clock synchronous mode with non-FIFO selected**

## (2) FIFO selected

Figure 30.66 shows an example of serial transmission in clock synchronous mode with FIFO selected.

In serial data transmission, the SCI operates as follows:

1. The SCI transfers data from the transmit-FIFO (TDR register) to the TSR register when data is written to transmit-FIFO (TDR register) in the SCIn\_TXI interrupt handling routine. The amount of data that can be written to transmit-FIFO (TDR register) is 16 - FTSR.T[5:0] bytes. In addition, when starting data transmission, set the CCR0.TIE bit and the CCR0.TE bit to 1 simultaneously by a single instruction. Then a SCIn\_TXI interrupt request is generated.
2. After transferring data from transmit-FIFO (TDR register) to TSR, the SCI starts transmission. When the amount of transmit data written in transmit-FIFO (TDR register) is equal to or less than the specified transmit triggering number, the CSR.TDRE is set to 1. When the CCR0.TIE bit is set to 1, an SCIn\_TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to transmit-FIFO (TDR register) in the SCIn\_TXI interrupt handling routine before transmission of the current transmit data has finished. When SCIn\_TEI interrupt requests are in use, set the CCR0.TIE bit to 0 and the CCR0.TEIE bit to 1 after the last of the data to be transmitted is written to the transmit-FIFO (TDR register) from the handling routine for SCIn\_TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when the clock output mode is specified and in synchronization with the input clock when the use of an external clock is specified. Output of the clock signal is suspended until the input CTS signal is low when the CCR1.CTSE bit is 1 (CTS function enabled).
4. The SCI checks whether non-transmitted data remains in transmit-FIFO (TDR register)<sup>\*1</sup> on output of the stop bit.
5. When data is remaining in the transmit-FIFO (TDR register), the next transmit data is transferred from transmit-FIFO (TDR register) to TSR and serial transmission of the next frame starts.

- If no data is remaining in the transmit FIFO (TDR register), the CSR.TEND flag is set to 1. The TXDn pin retains the output state of the last bit. If the CCR0.TEIE bit is 1, an SCIn\_TEI interrupt request is generated and the SCKn pin is held high.

Note 1. The number of unsent transmit data stored in the TDR register (transmit FIFO) can be monitored by reading the FTSR.T[5:0] bits.

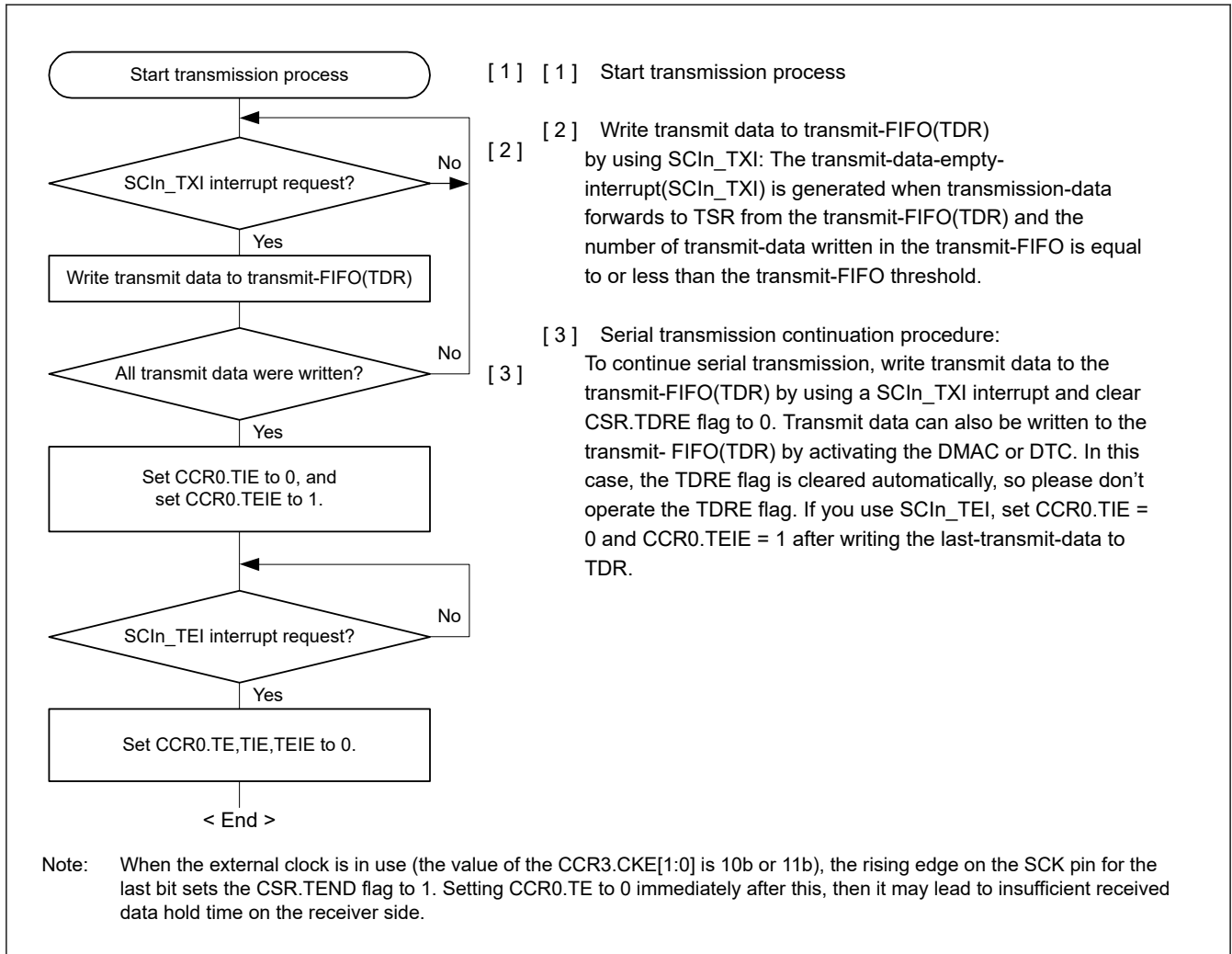


Figure 30.66 Example flow of serial transmission in clock synchronous mode with FIFO selected

### 30.6.5 Serial Data Reception in Clock Synchronous Mode

#### (1) Non-FIFO selected

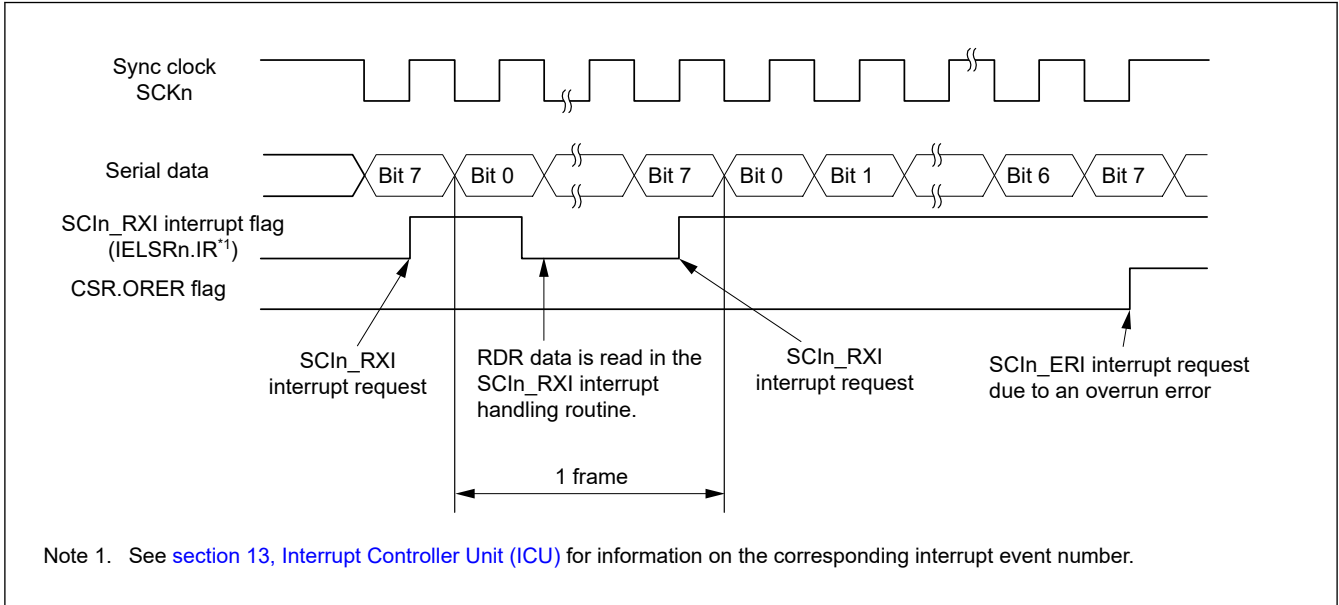
Figure 30.67 and Figure 30.68 show examples of SCI operation for serial reception in clock synchronous mode.

In serial data reception, the SCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

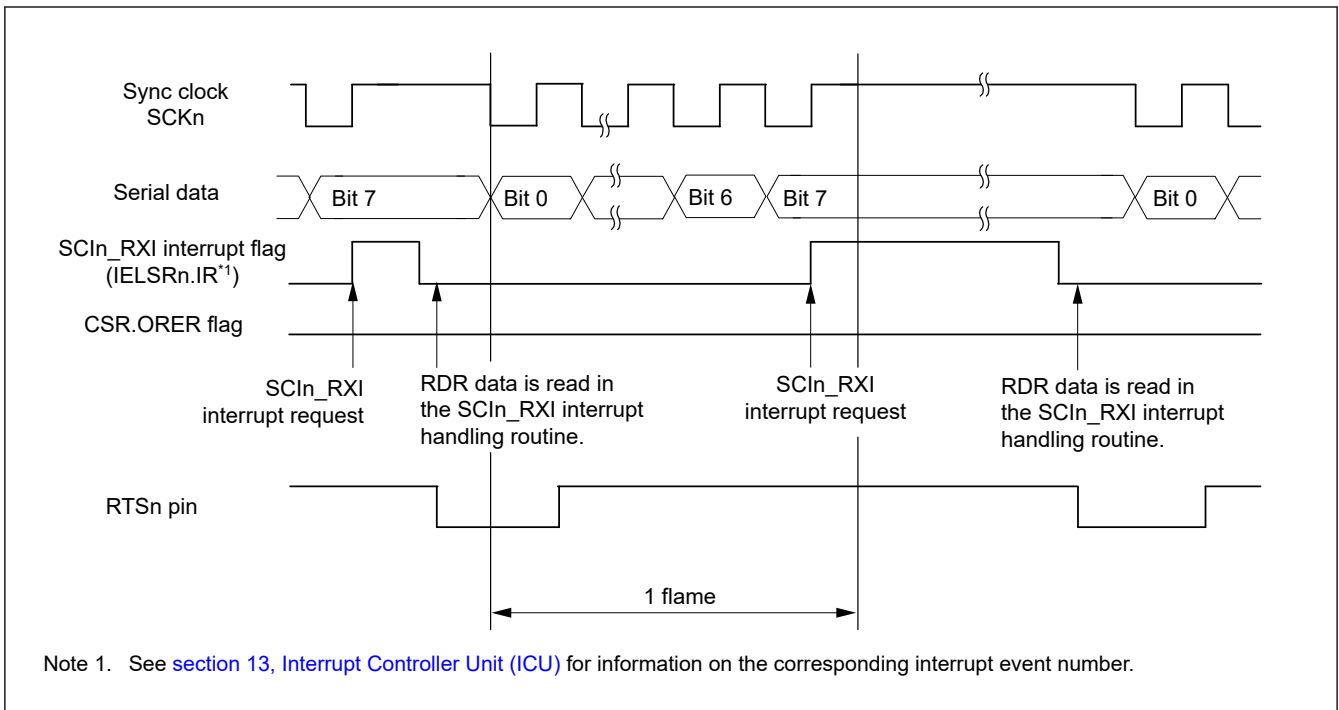
- When the value of the CCR0.RE bit becomes 1, the CTSn\_RTSn pin goes low (when the RTS function is used).
- The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
- If an overrun error occurs, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn\_ERI interrupt request is generated. Receive data is not transferred to the RDR register.
- When reception completes successfully, receive data is transferred to the RDR register. If the CCR0.RIE bit is 1, an SCIn\_RXI interrupt request is generated. Continuous reception is enabled by reading the received data transferred to the

RDR register in the SCIn\_RXI interrupt handling routine before reception of the next receive data completes. Reading the received data that is transferred to RDR causes the CTSn\_RTSn pin to output low (when the RTS function is used).

If you want to prevent the CTSn\_RTSn pin output from turning low level after the final data is received, clear the CCR0.RE bit to 0 and then read the RDR register.



**Figure 30.67 Example operation for serial reception in clock synchronous mode (1) when the RTS function is not used**



**Figure 30.68 Example operation for serial reception in clock synchronous mode (2) when RTS function is used**

Data transfer cannot resume while the receive error flag is 1. Therefore, clear the ORER, FER, and PER flags in the CSR register to 0 before resuming data reception. Additionally, always read the RDR register during overrun error processing. When a data reception is forced to terminate by a 0 write to the CCR0.RE bit during operation, read the RDR register because received data that is not yet read might be left in the RDR register.

[Figure 30.69](#) shows an example flow of serial data reception.

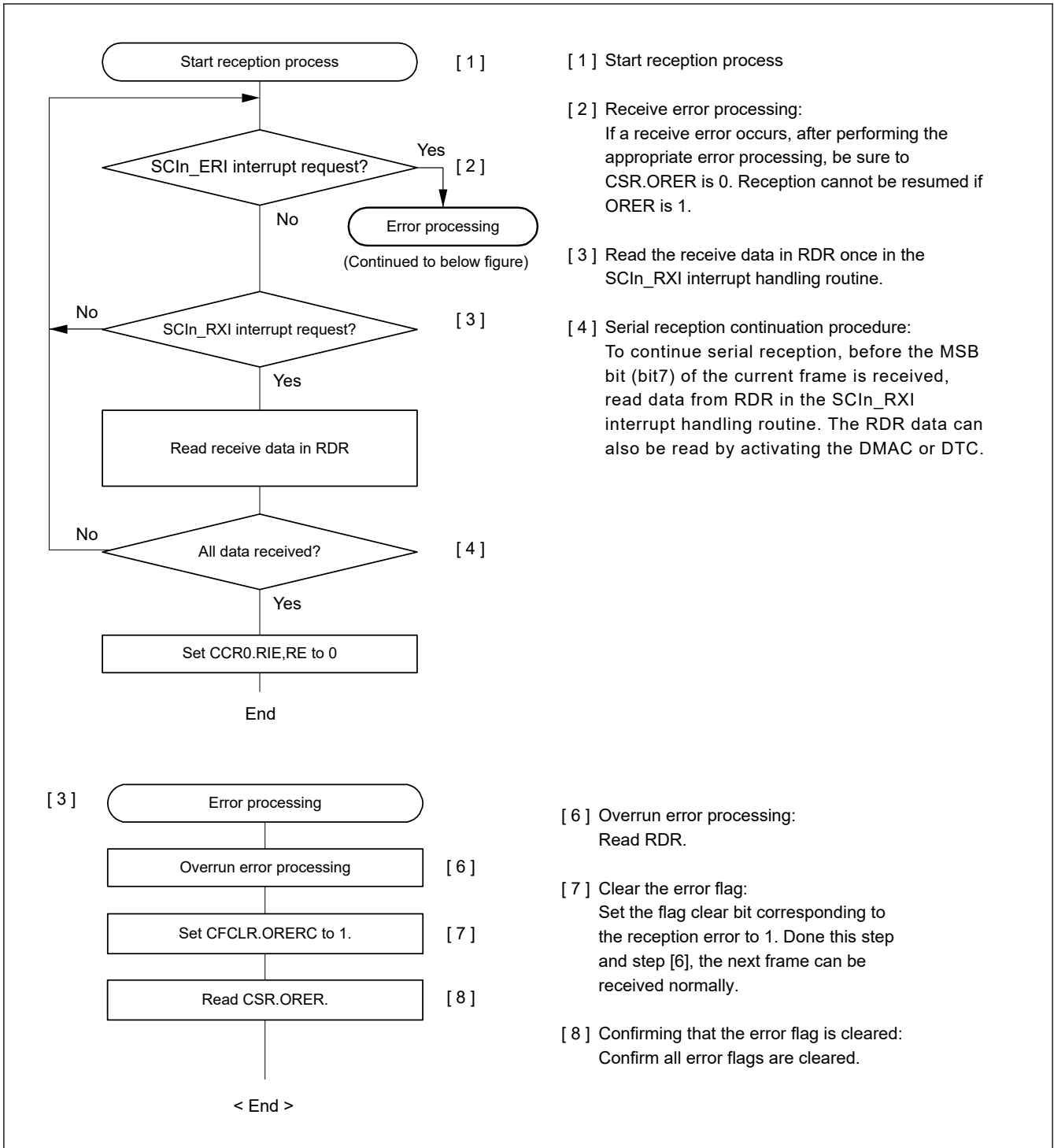


Figure 30.69 Example flow of serial reception in clock synchronous mode with non-FIFO selected

(2) FIFO selected

Figure 30.70 shows an example of serial reception in clock synchronous mode with FIFO selected.

In serial data reception, the SCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the value of the CCR0.RE bit becomes 1, the CTSn\_RTSn pin goes low (when the RTS function is used).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and transfers the received data to the receive-FIFO (RDR register).
3. If an overrun error occurs, the CSR.ORER flag is set to 1. If the CCR0.RIE bit is 1, an SCIn\_ERI interrupt request is generated. Received data is not transferred to the receive-FIFO (RDR register)\*1.



4. When data reception completes successfully, the receive data is transferred to the receive-FIFO (RDR register)<sup>\*1</sup>. The CSR.RDRF flag is set to 1 when the amount of the receive data stored in receive-FIFO (RDR register) is equal to or greater than the specified receive triggering number. If the CCR0.RIE bit is 1, an SCIn\_RXI interrupt request is generated. Continuous data reception is enabled by reading the receive data transferred to receive-FIFO (RDR register)<sup>\*1</sup> in the SCIn\_RXI interrupt handling routine before an overrun error occurs. If the amount of received data that is transferred to receive-FIFO (RDR register) is less than the specified receive triggering number, the CTSn\_RTSn pin goes low (when the RTS function is used).

Note 1. In clock synchronous mode, RDR.RDAT[8] is not used.

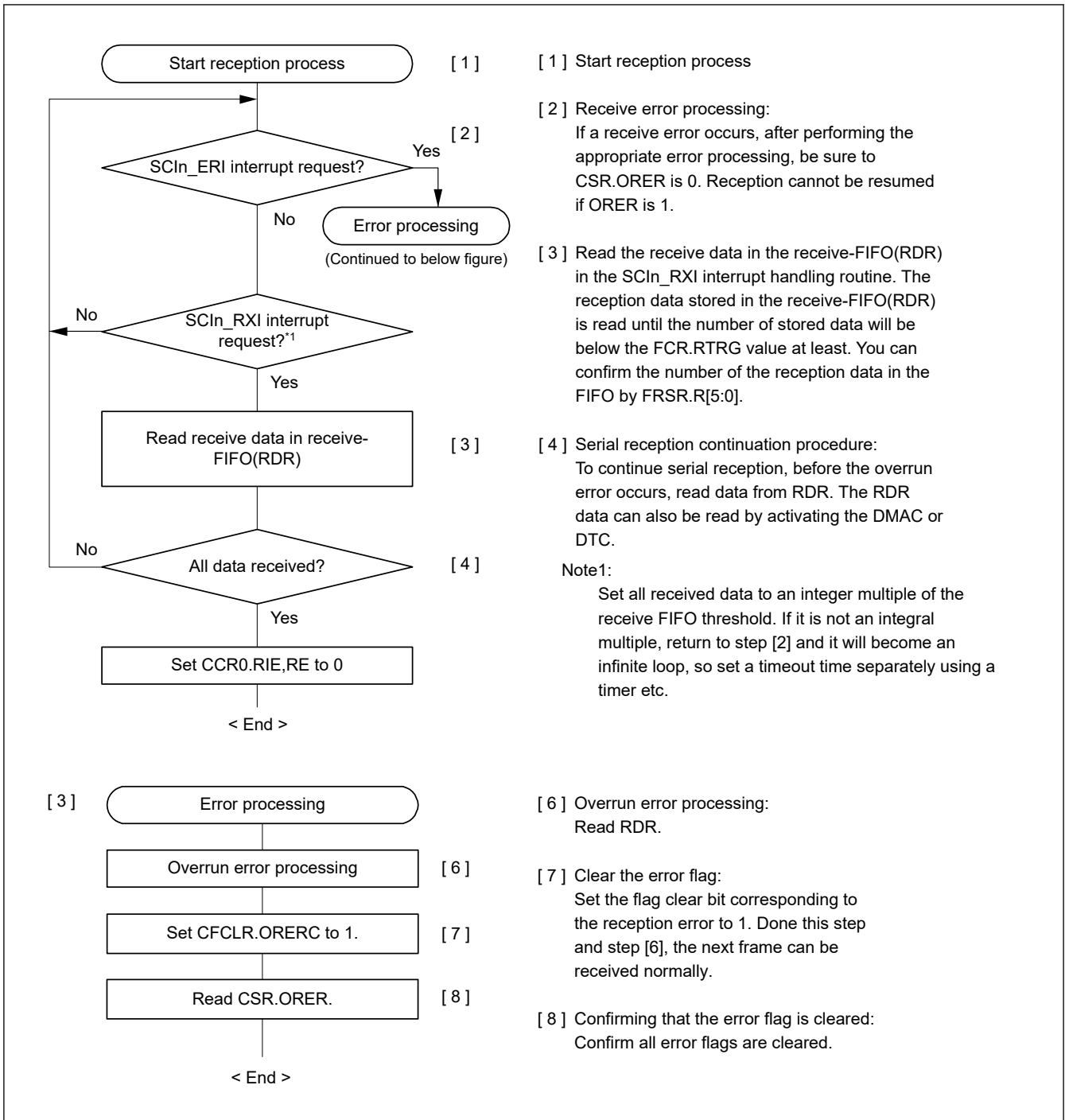


Figure 30.70 Example flow of serial reception in clock synchronous mode with FIFO selected

### 30.6.6 Simultaneous Serial Data Transmission and Reception in Clock Synchronous Mode

#### (1) Non-FIFO selected

Figure 30.71 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode. After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

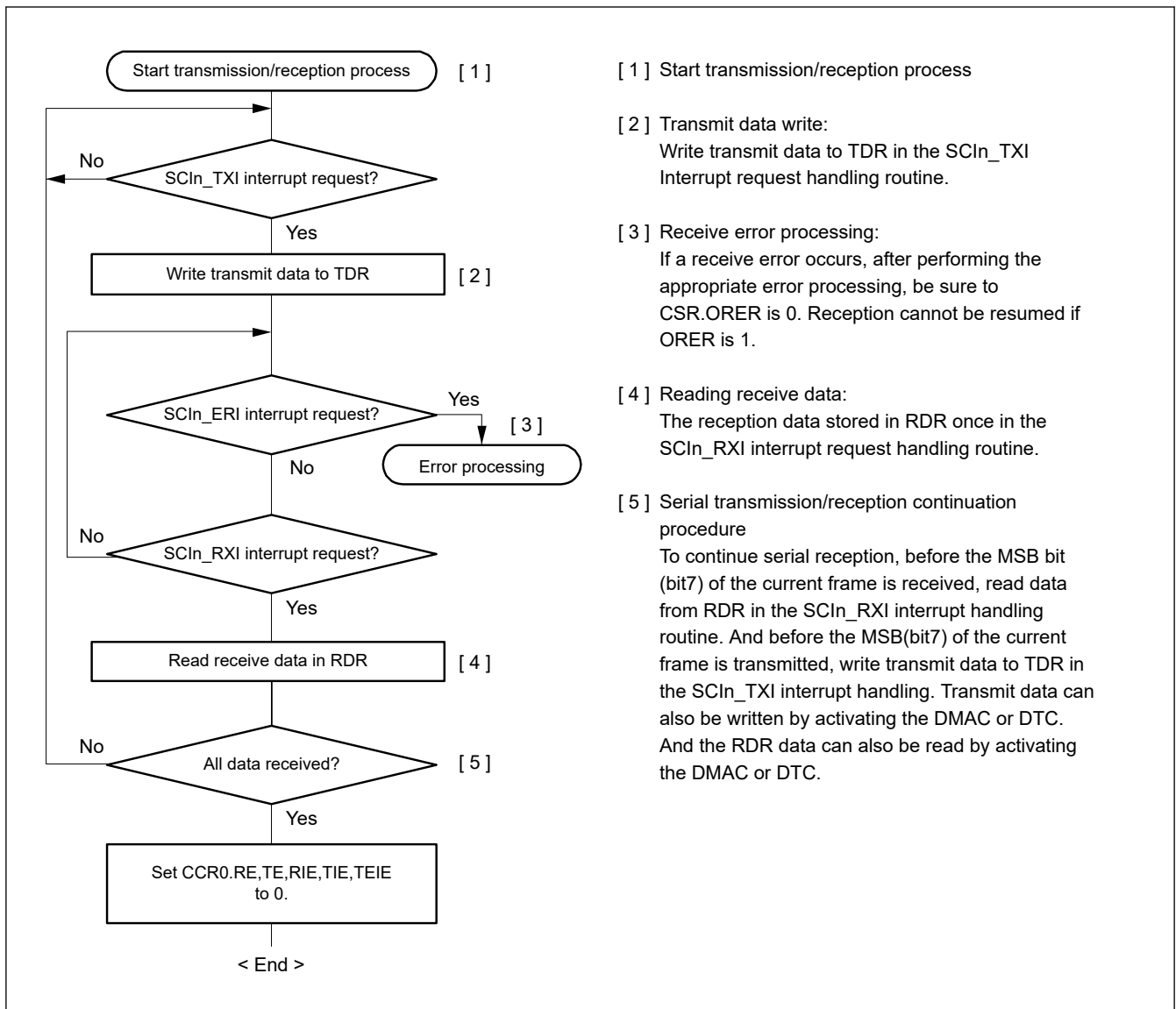
To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data transmission by verifying that the CSR.TEND flag is set to 1.
2. Initialize the CCR0 register, and then set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the data reception.
2. Set the CCR0.TE and RE bits to 0, and then check that the receive error flag (ORER, FER, and PER) in the CSR register is 0.
3. Set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

When the RTS function is used in the concurrent transmission/reception operation, if you want to prevent the CTSn\_RTSn pin output from turning to low after the final data is received as in the reception operation, clear the RE and TE bits in CCR0 to 0 simultaneously, and then read the RDR register.



**Figure 30.71 Example flow of simultaneous serial transmission and reception in clock synchronous mode with non-FIFO selected**

## (2) FIFO selected

Figure 30.72 shows an example flow of simultaneous serial transmit and receive operations in clock synchronous mode with FIFO selected.

After initializing the SCI, use the following procedure for simultaneous serial data transmit and receive operations.

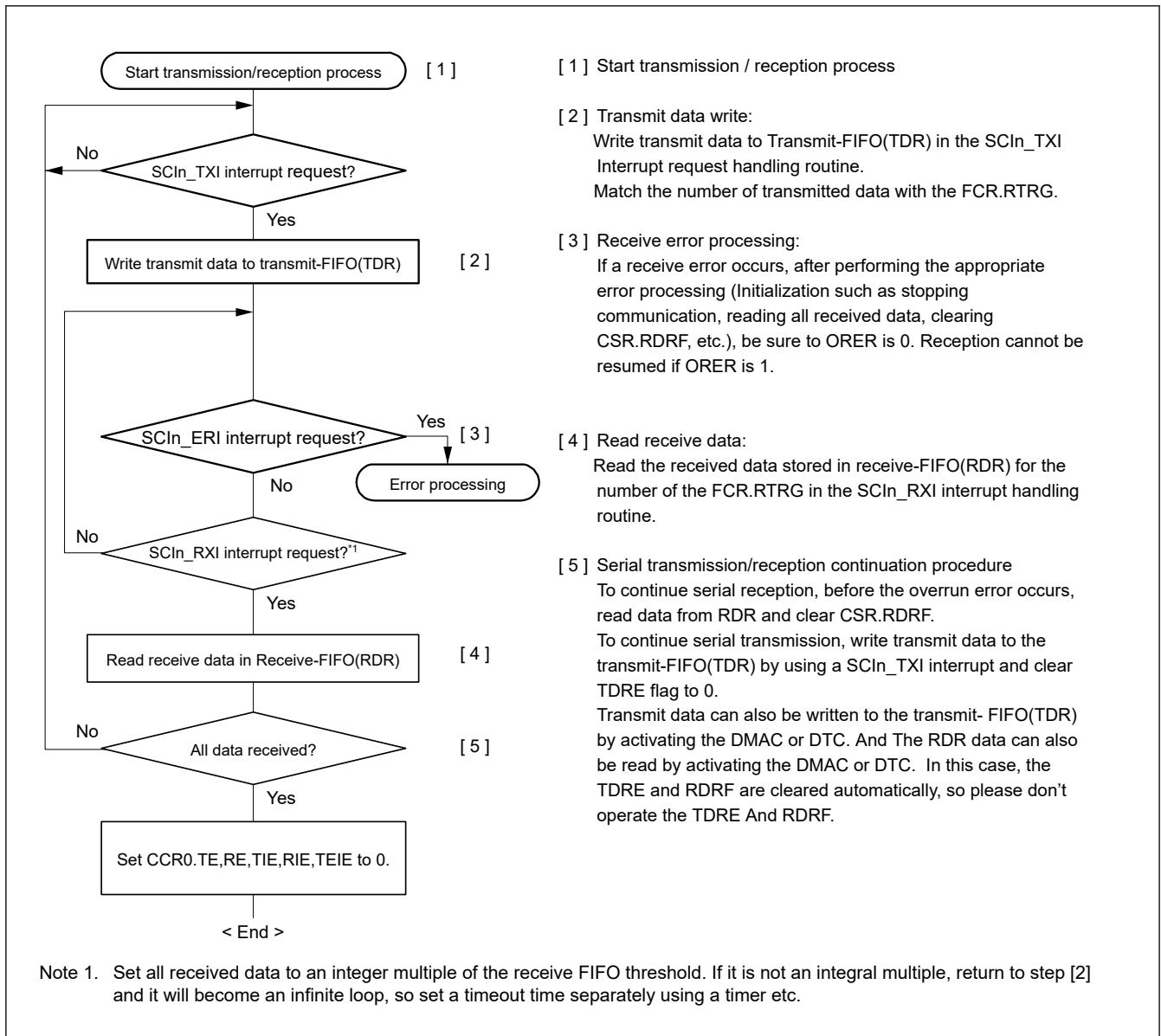
To switch from transmit mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the transmission by verifying that the CSR.TEND flag is set to 1.
2. Initialize the CCR0 register, then set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode:

1. Check that the SCI completes the reception.
2. Set the CCR0.TE and RE bits to 0.
3. Check that the receive error flags (ORER, FER, and PER) in the CSR register are 0, and then set the TIE, RIE, TE, and RE bits in the CCR0 register to 1 simultaneously by a single instruction.

Since clock synchronous communication performs transmission and reception at the same time, make sure that the number of data to be transmitted and received is the same.



**Figure 30.72 Example flow of simultaneous serial transmission and reception in clock synchronous mode with FIFO selected**

### 30.6.7 Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used

When the clock synchronous internal clock is used (master mode), the clock enabled by CCR4.SCKSEL bit is used as a reception sampling clock.

This function adjusts the reception sampling timing by delaying the clock by 1 to 4 TCLK and adding a digital delay.

Setting the CCR4.ASEN bit to 1 enables this function. The delay value is set in CCR4.AST[1:0].

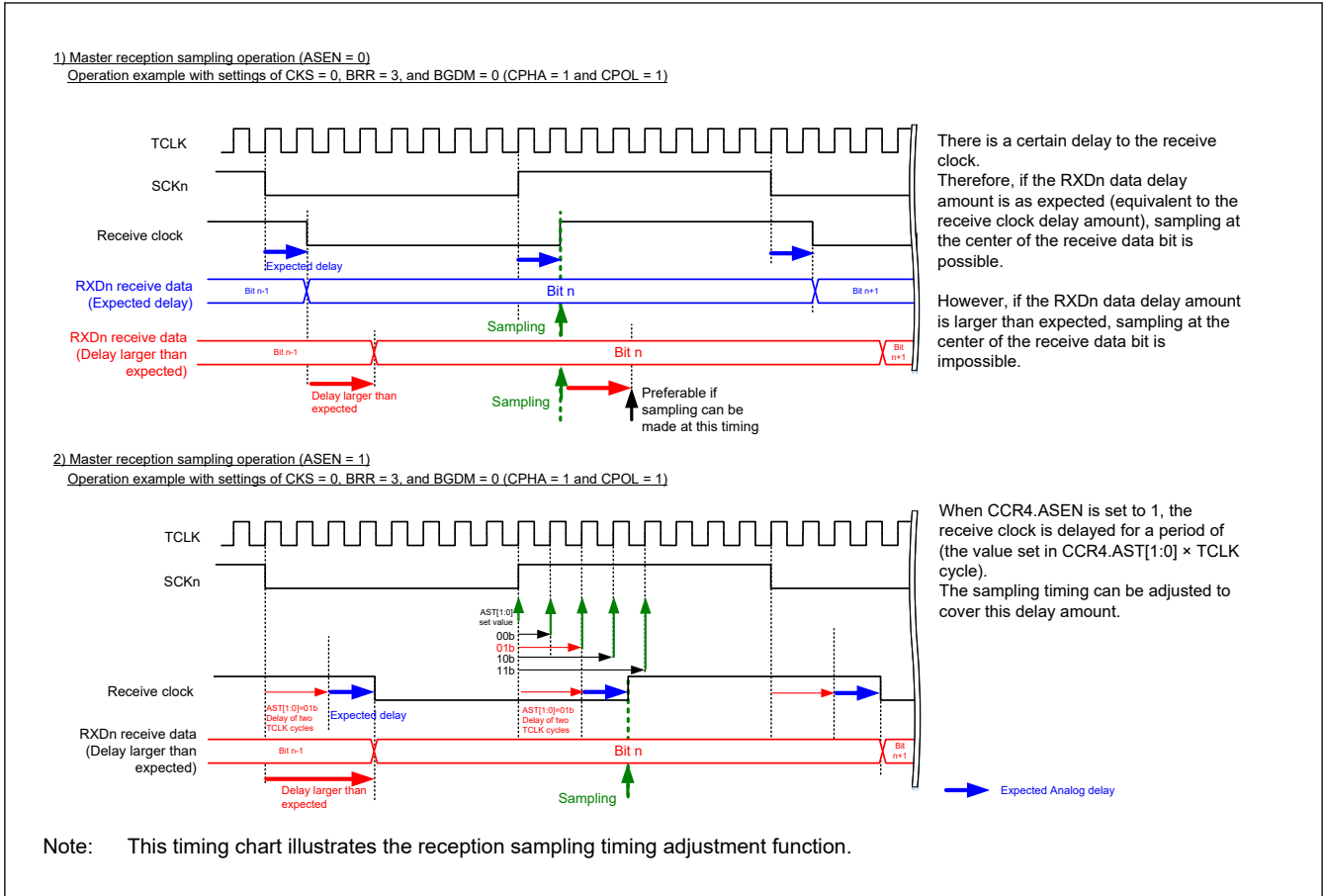


Figure 30.73 Reception Sampling Timing Adjustment Operation in Clock Synchronous Mode (Master)

### 30.7 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.

Smart card interface mode can be selected using the appropriate register.

#### 30.7.1 Example Connection

Figure 30.74 shows an example connection between a smart card (IC card) and the MCU. As shown in Figure 30.74, because the MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.

Setting the CCR0.TE and CCR0.RE bits to 1 with an IC card disconnected enables closed-loop transmission or reception, allowing self-diagnosis. To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.

An output port of the MCU can be used to output a reset signal.

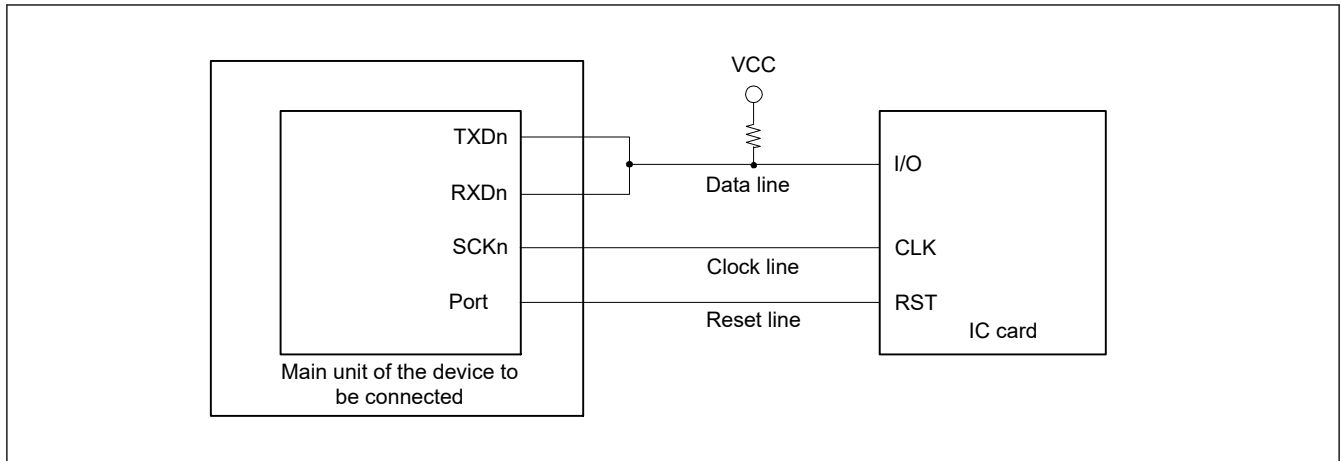


Figure 30.74 Example connection with a smart card (IC card)

### 30.7.2 Data Format (Except in Block Transfer Mode)

Figure 30.75 shows the data transfer formats in smart card interface mode:

- One frame consists of 8-bit data and a parity bit in Asynchronous mode.
- During transmission, at least 2 etus (elementary time unit – the time required for transferring 1 bit) is set as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low error signal is output for 1 etu after 10.5 etus elapse from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etus.

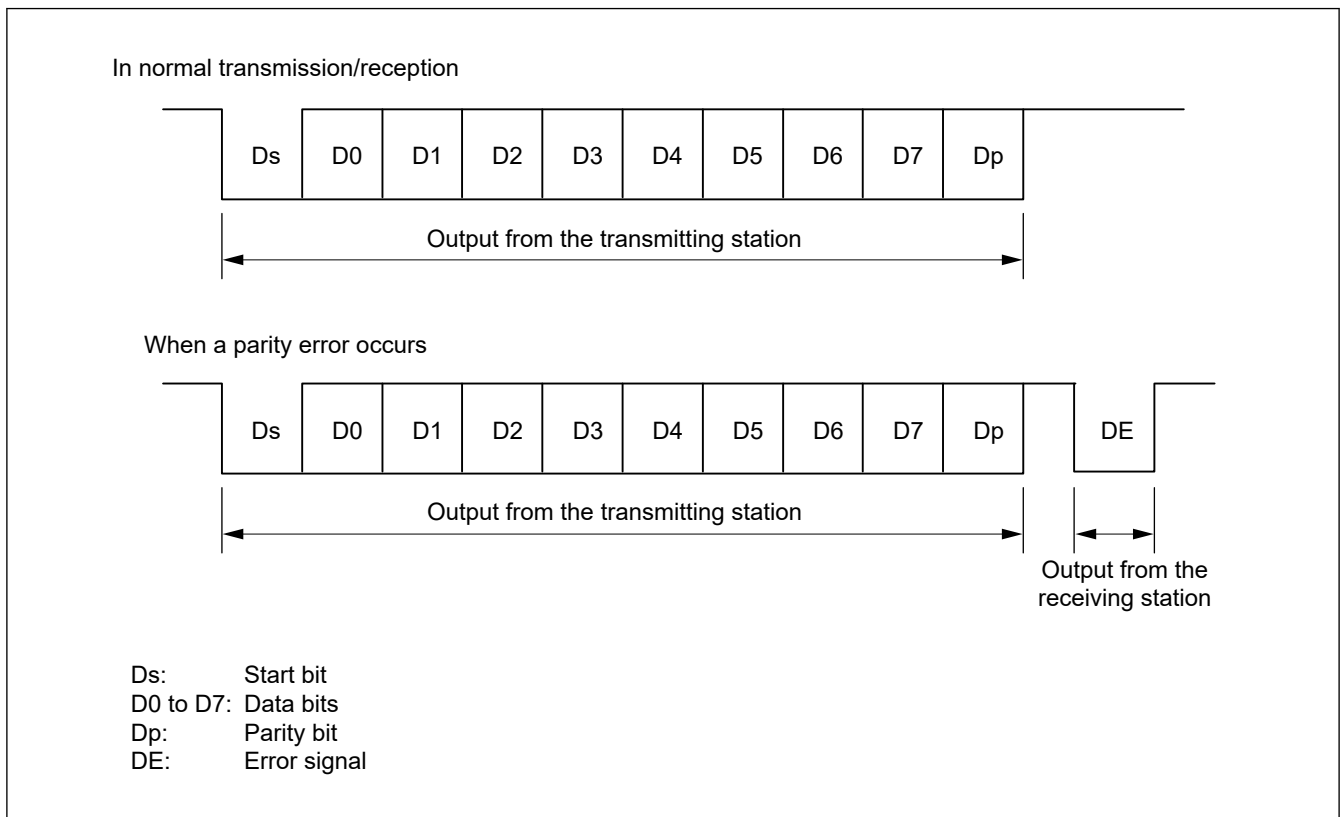


Figure 30.75 Data formats in smart card interface mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedures in this section.

### (1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 indicate the Z and A states, respectively, and data is transferred with LSB-first for the start character, as shown in Figure 30.76. Therefore, data in the start character in the figure is 0x3B.

When using the direct convention type, write 1 to the CCR3.LSBF and write 0 to the CCR3.SINV. Write 0 to the CCR1.PM bit to use even parity, which is prescribed by the smart card standard.

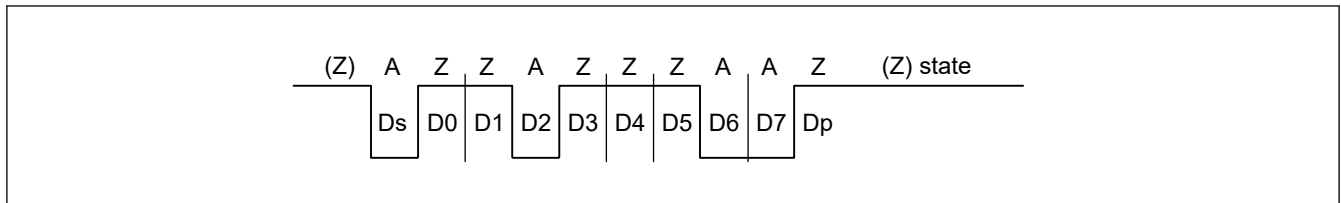


Figure 30.76 Direct convention with LSBF in CCR3 = 1, SINV in CCR3 = 0, and PM in CCR1 = 0

### (2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 indicate the A and Z states, respectively, and data is transferred with MSB-first for the start character, as shown in Figure 30.77. Therefore, data in the start character in the figure is 0x3F.

When using the inverse convention type, write 0 to the CCR3.LSBF and write 1 to the CCR3.SINV. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to the Z state. Because the SINV bit of the MCU only inverts data bits D7 to D0, write 1 to the PM bit in CCR1 to invert the parity bit for both transmission and reception.

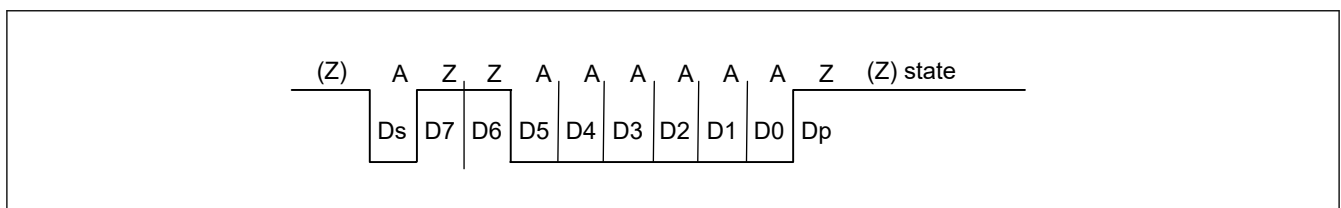


Figure 30.77 Inverse convention with LSBF in CCR3 = 0, SINV in CCR3 = 1, and PM in CCR1 = 1

## 30.7.3 Block Transfer Mode

Block transfer mode differs from normal smart card interface mode as follows:

- Even if a parity error is detected during reception, no error signal is output. Because the PER flag in CSR is set by error detection, clear the PER flag before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is set as a guard time from the end of the parity bit until the start of the next frame
- Because the same data is not retransmitted, the TEND flag in CSR is set to 11.5 etus after transmission starts
- In block transfer mode, the ERS flag in CSR indicates the error signal status as in normal smart card interface mode, but the flag is read as 0 because no error signal is transferred

## 30.7.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate set up in the CCR2.BCP[2:0] bits. The frequency is always 16 times the bit rate in normal Asynchronous mode.

For data reception, the falling edge of the start bit is sampled with the base clock to perform internal synchronization.

Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 30.78. The reception margin is determined by the following formula:

$$M = \left| \left( 0.5 - \frac{1}{2N} \right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100 \text{ [%]}$$

M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, 256)

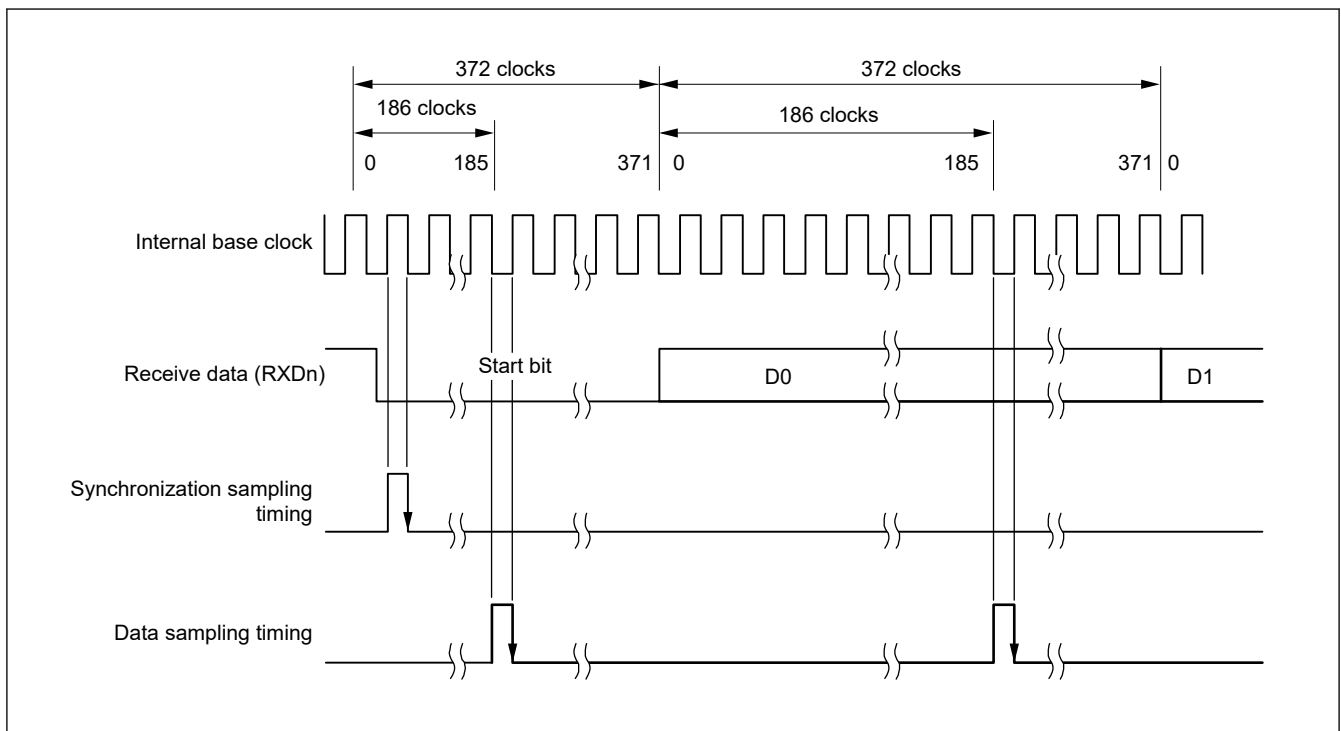
D: Duty cycle of clock (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the specified formula, the reception margin is determined using the following formula:

$$M = \{0.5 - 1/(2 \times 372)\} \times 100 \text{ [%]} = 49.866 \text{ %}$$



**Figure 30.78** Receive data sampling timing in smart card interface mode when the clock frequency is 372 times the bit rate

### 30.7.5 SCI Initialization (Smart Card Interface Mode)

Before transmitting and receiving data, write the initial value 0x00 in the CCR0 register and initialize the SCI following the example flow shown in [Table 30.40](#).

Always set the initial value in the TIE, RIE, TE, RE, TEIE bits in the CCR0 register before switching from transmission to reception mode or from reception to transmission mode. When CCR0.RE is set to 0, the RDR register is not initialized.

In transmission mode, set 1 to the CCR0.TE bit and CCR0.TIE bit simultaneously, then the SCIn\_TXI interrupt request is generated.

To change from reception mode to transmission mode, first check that reception has completed, then initialize the SCI. At the end of initialization, set CCR0.TE = 1 and CCR0.RE = 0. Reception completion can be verified by reading the SCIn\_RXI request, ORER, or PER flag in CSR.

To change transmission mode to reception mode, first check that transmission has completed, then initialize the SCI. At the end of initialization, set CCR0.TE = 0 and CCR0.RE = 1. Transmission completion can be verified by reading the TEND flag in CSR.

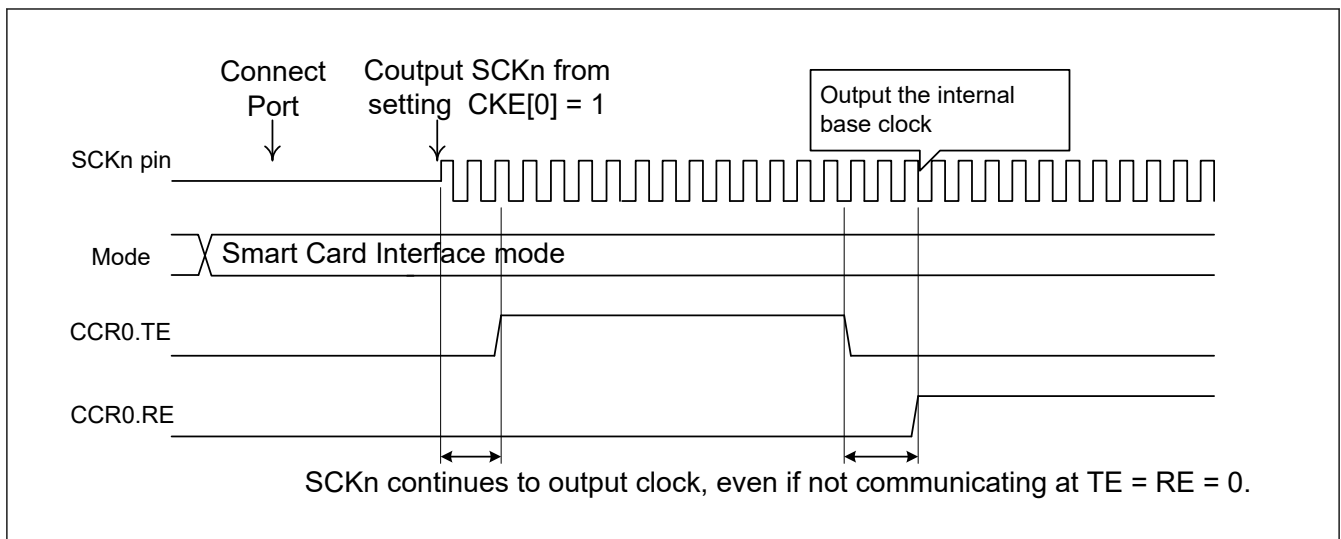


**Table 30.40 Example flow of SCI initialization in smart card interface mode**

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set CCR3	Set communication mode (MOD[2:0] = 001b), BLK, GM, and SINV. Leave other bits at their initial values.
4	Set CCR2	Set clock-select and bit-rate. Set BRME to 0.
5	Set CCR1	Set up the Noise filter function, communication terminal status. Set NFEN, PE, CTSE to 0 and set PE to 1.
6	Set the I/O port functions	Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
7	Set CCR3	Set CKE[1:0]. When the CKE[0] bit is set to "1" due on GM setting value, the clock is output from the SCKn pin.
8	Set CFCLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC,FERC,PERC,MFFC,ORERC,DFERC,DPERC,DCMFC,ERSC
9	Set CCR0	Set the TE or RE to 1. And set the TIE and RIE. Do not simultaneously set the TE and RE bits to 1 if self-diagnosis in not used.
10	Initialization completed	—

Figure 30.79 is a timing chart when data transmission is performed by making transition to the Smart Card Interface mode according to the above flow chart. The figure shows the case when CCR3.GM bit is 0. As shown in the figure, when the pin function is set to the SCKn pin, the SCKn pin is high impedance because the CCR3.CKE [0] bit is 0. When the TXDn pin is set, the TXDn pin is high impedance because the CCR0.TE bit is 0. Start clock output to the SCKn pin with the clock output setting CCR3.CKE [0] to 1, start data transmission by writing transmit data after setting CCR0.TE to 1.

In the smart card interface mode, even if not communicating at CCR0.TE = 0 and CCR0.RE = 0, the clock is continuously output if the clock output setting is used.



**Figure 30.79 Example of Timing chart of data transmission in Smart Card Interface Mode**

### 30.7.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode) is different from that in non-smart card interface mode, in that an error signal is sampled and data can be re-transmitted in smart card mode. Figure 30.80 shows the data re-transfer operation during transmission.

1. When an error signal from the receiver end is sampled after 1-frame data is transmitted, the CSR.ERS flag is set to 1. If the CCR0.RIE bit is 1, an SCIn\_ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.

2. For a frame in which an error signal is received, the CSR.TEND flag is not set. Data is re-transferred from TDR to TSR, allowing automatic data retransmission.
3. If no error signal is returned from the receiver, the ERS flag is not set to 1.
4. In this case, the SCI determines that transmission of 1-frame data, including the re-transfer, is complete, and the TEND flag is set. If the CCR0.TIE bit is 1, an SCIn\_TXI interrupt request is generated. Write transmit data to the TDR to start transmission of the next data.

Figure 30.82 shows an example flow of serial transmission. All the processing steps are automatically performed using an SCIn\_TXI interrupt request to activate the DTC or DMAC.

When the CSR.TEND flag is set to 1 in transmission and when the CCR0.TIE bit is 1, an SCIn\_TXI interrupt request is generated.

The DTC or DMAC is activated by an SCIn\_TXI interrupt request if the SCIn\_TXI interrupt request is previously specified as a source of DTC or DMAC activation, allowing the transfer of transmit data. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept at 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission when an error occurs. Because the ERS flag is not automatically cleared, set the RIE bit to 1 before enabling an SCIn\_ERI interrupt request to be generated if an error occurs, and clear the ERS flag to 0.

When transmitting or receiving data using the DTC or DMAC, always enable the DTC or DMAC before making the SCI settings.

For DTC or DMAC settings, see section 17, Data Transfer Controller (DTC), section 16, DMA Controller (DMAC).

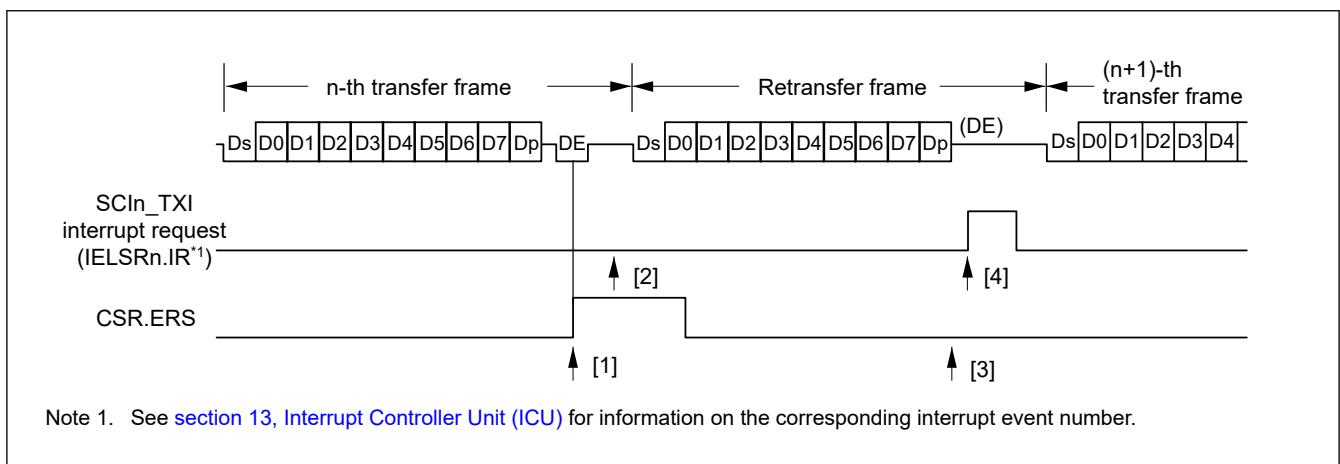


Figure 30.80 Data re-transfer operation in smart card interface transmission mode

The CSR.TEND flag is set at different timings depending on the CCR3.GM bit setting. Figure 30.81 shows the TEND flag generation timing.

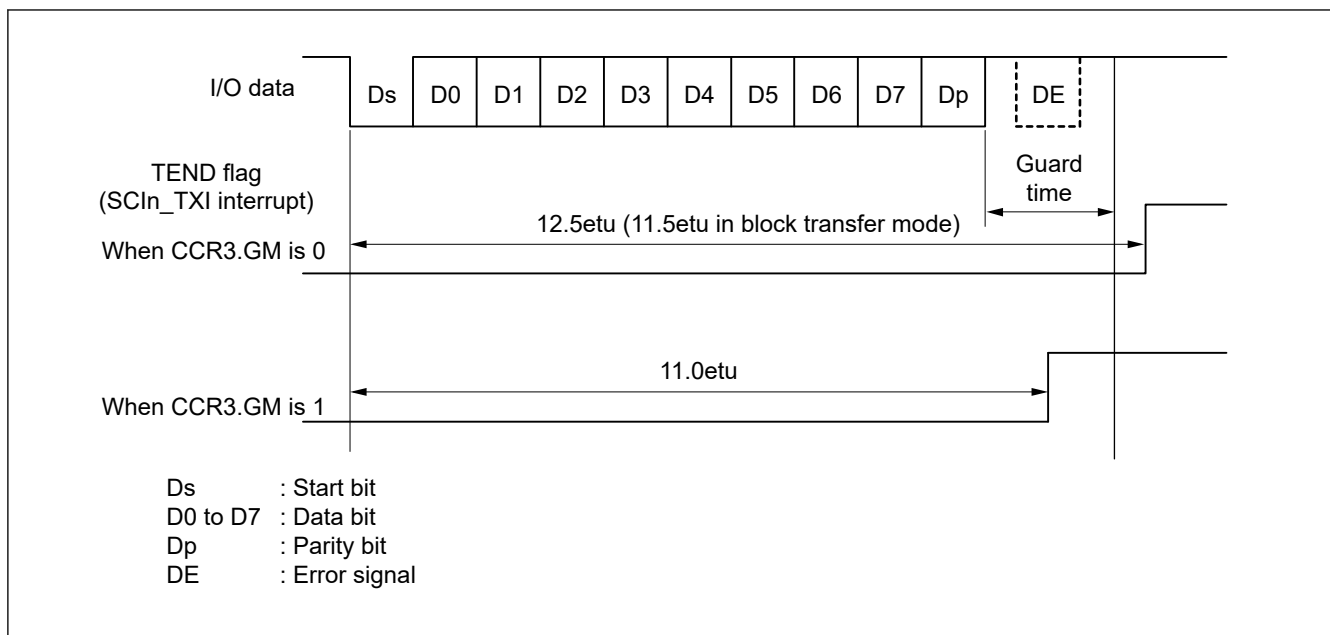
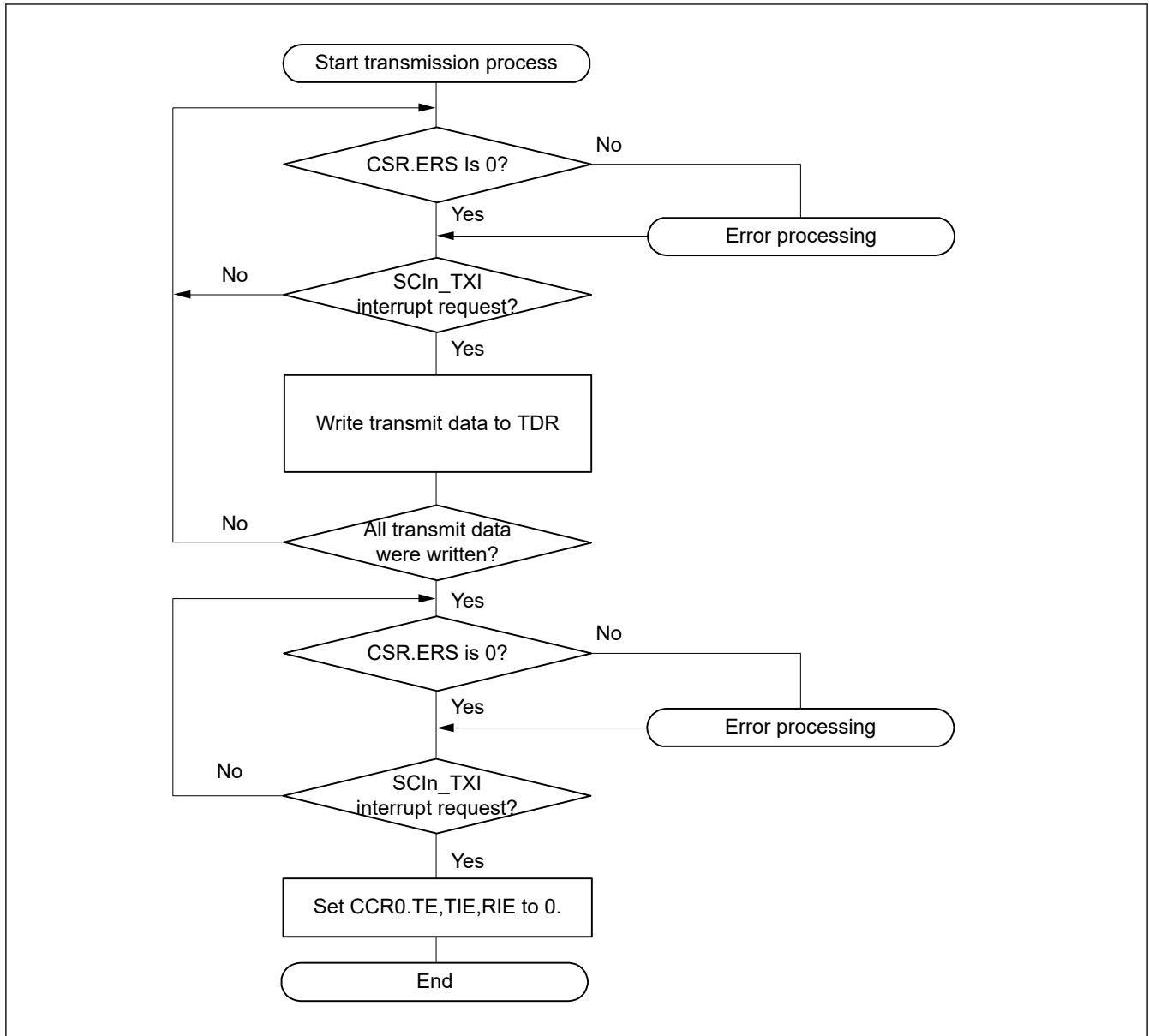


Figure 30.81 CSR.TEND flag generation timing during transmission



**Figure 30.82 Example flow of smart card interface transmission**

### 30.7.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. [Figure 30.83](#) shows the data re-transfer operation in reception mode.

1. If a parity error is detected in the receive data, the CSR.PER flag is set to 1. When the CCR0.RIE bit is 1, an SCIn\_ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no SCIn\_RXI interrupt is generated.
3. When no parity error is detected, the CSR.PER flag is not set to 1.
4. In this case, data is determined to be received successfully. When the CCR0.RIE bit is 1, an SCIn\_RXI interrupt request is generated.

[Figure 30.84](#) shows an example flow of serial data reception. All the processing steps are automatically performed using an SCIn\_RXI interrupt request to activate the DTC or DMAC.

In reception, setting the RIE bit to 1 allows an SCIn\_RXI interrupt request to be generated. The DTC or DMAC is activated by an SCIn\_RXI interrupt request if the SCIn\_RXI interrupt request is previously specified as a source of DTC or DMAC activation, allowing the transfer of receive data.

If an error occurs during reception and either the ORER or PER flag in CSR is set to 1, a receive error interrupt (SCIn\_ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DTC or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred.

If a parity error occurs and the PER flag is set to 1 during reception, the receive data is transferred to RDR, allowing the data to be read.

When a reception is forced to terminate by setting CCR0.RE to 0 during operation, read the RDR register because the received data that is not yet read might be left in the RDR.

Note: For operations in block transfer mode, see [section 30.3.9. Serial Data Reception in Asynchronous Mode](#).

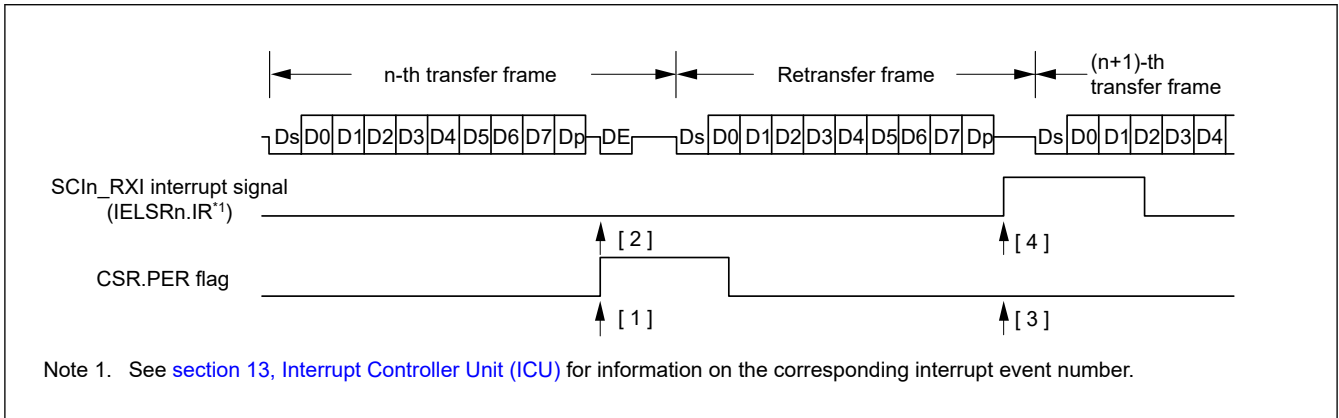


Figure 30.83 Data re-transfer operation in smart card interface reception mode

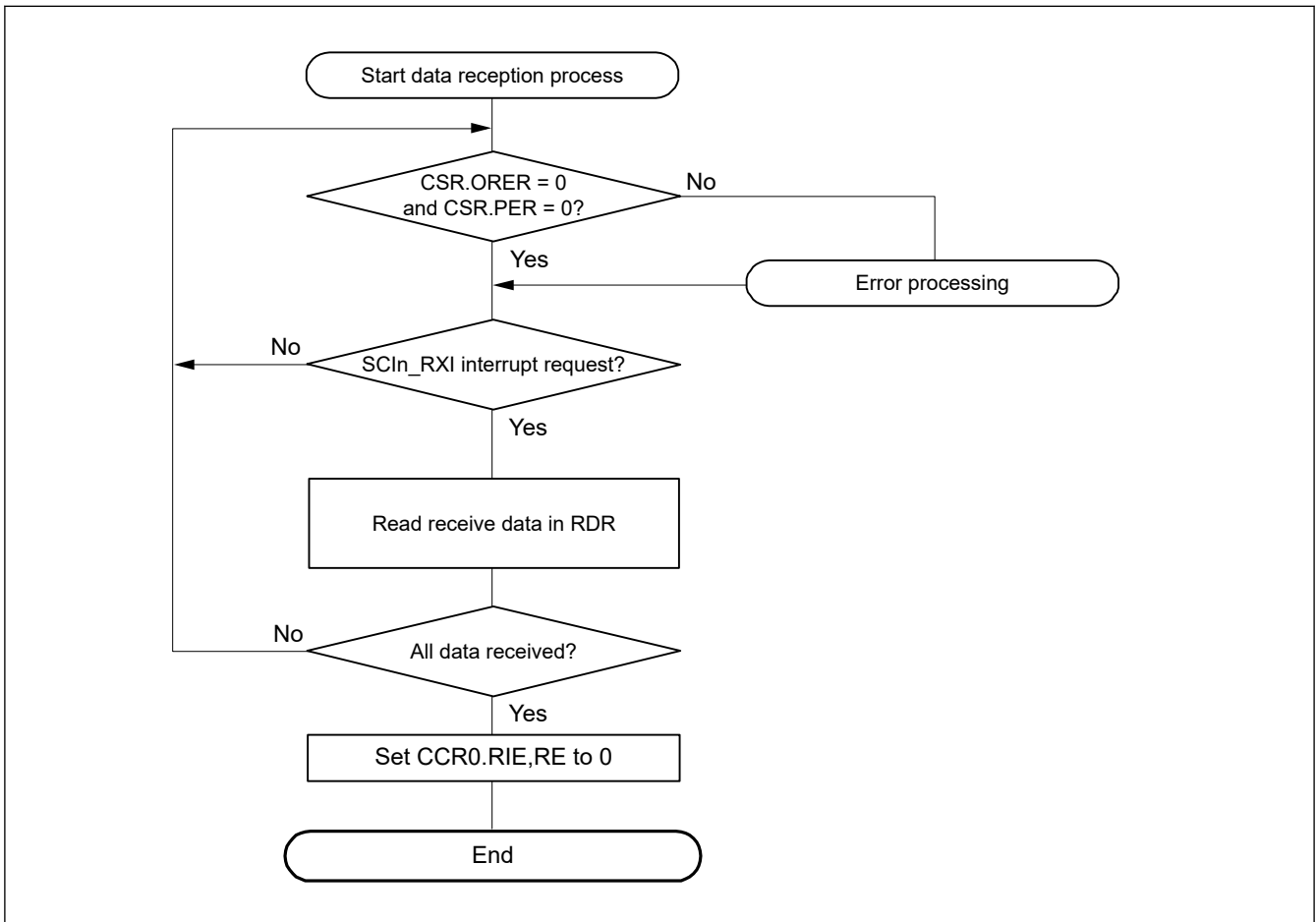


Figure 30.84 Example flow of smart card interface reception

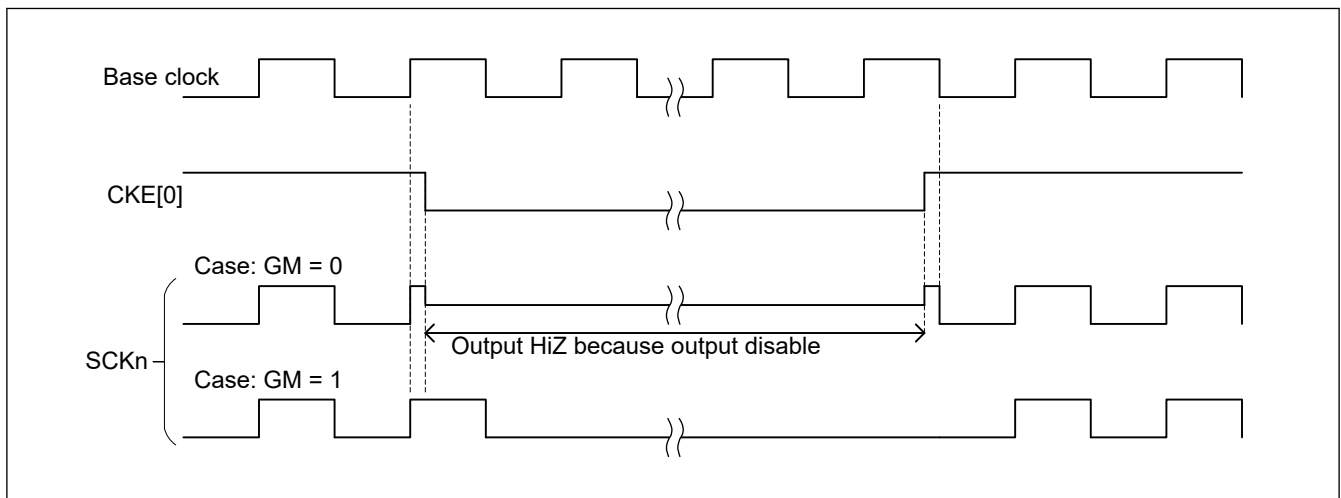
### 30.7.8 Clock Output Control

When the GM bit in CCR3 is set to 1, the clock output can be controlled by the CKE[1:0] bits in CCR3. For details on the CKE[1:0] bits, see [section 30.2.8. CCR3 : Common Control Register 3](#). When setting the clock output, the base clock described in [section 30.7.4. Receive Data Sampling Timing and Reception Margin](#) the bit rate is set by CCR2.CKS, CCR2.BCP[2:0] and BRR[7:0].

[Figure 30.85](#) shows an example timing for the clock output control when the CKE[1] bit in CCR3 is set to 0 and the CKE[0] bit in CCR3 is controlled.

When the GM bit in CCR3 is 0, output control by the CKE[0] bit in CCR3 is immediately reflected on the SCKn pin, so there is a possibility that pulses with an unintended width may be output from the SCKn pin.

When the GM bit in CCR3 is 1, the output pulse control by the CCR3.CKE [0] controls the pulse width set to be based on the state of the base clock.



**Figure 30.85** Clock Output timing

## 30.8 Operation in Simple IIC Mode

Simple IIC mode format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device can specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8 data bits in all frames are transmitted in order from the MSB.

The I<sup>2</sup>C bus format and timing of the I<sup>2</sup>C bus are shown in [Figure 30.86](#) and [Figure 30.87](#).

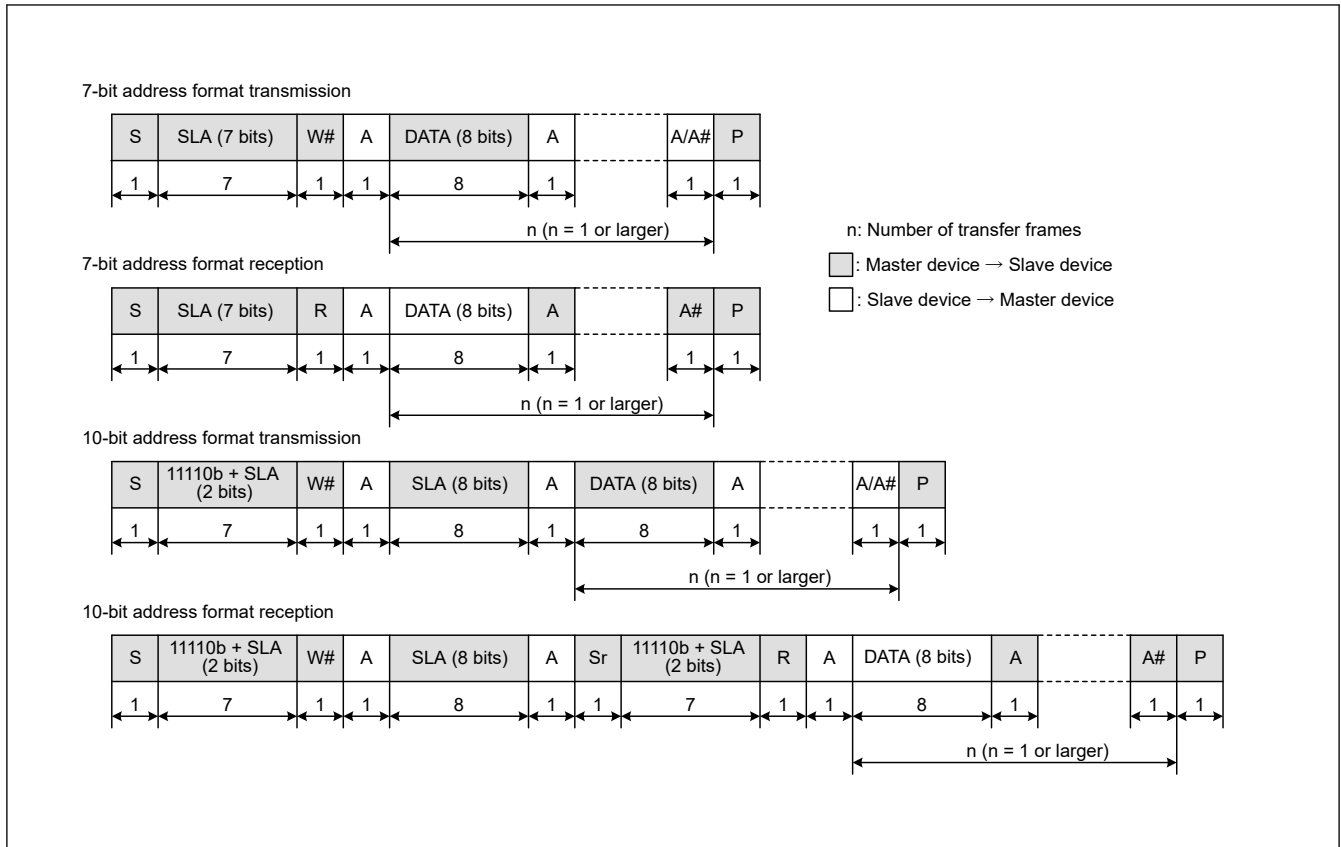


Figure 30.86 I<sup>2</sup>C bus format

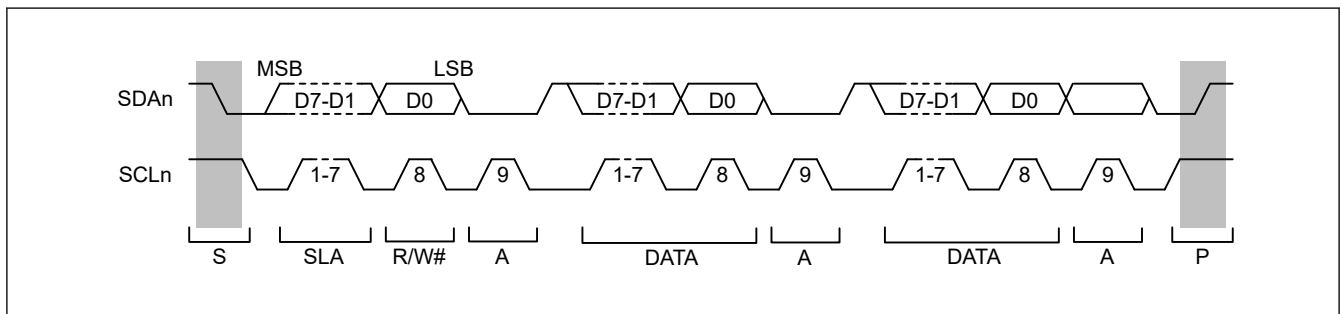


Figure 30.87 I<sup>2</sup>C bus timing when SLA is 7 bits

- **S:** Indicates a start condition, when the master device changes the level on the SDA<sub>n</sub> line from high to low while the SCL<sub>n</sub> line is high
- **SLA:** Indicates a slave address, by which the master device selects a slave device
- **R/W#:** Indicates the direction of transfer (reception or transmission). The value 1 indicates transfer from the slave device to the master device and 0 indicates transfer from the master device to the slave device.
- **A/A#:** Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return low indicates ACK and return high indicates NACK.
- **Sr:** Indicates a restart condition, when the master device changes the level on the SDA<sub>n</sub> line from high to low while the SCL<sub>n</sub> line is high and after the setup time elapses
- **DATA:** Indicates the data being received or transmitted
- **P:** Indicates a stop condition, when the master device changes the level on the SDA<sub>n</sub> line from low to high while the SCL<sub>n</sub> line is high

### 30.8.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the ICR.IICSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations:

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept in the released state
- The hold time for the start condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The level on the SCLn line falls (from the high level to the low level), the IICSTAREQ bit in ICR is set to 0, and a start-condition generated interrupt is output

Writing 1 to the IICRSTAREQ bit in ICR causes the generation of a restart condition. The generation of a restart condition proceeds through the following operations:

- The SDAn line is released and the SCLn line is kept at the low level
- The period at low level for the SCLn line is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The SCLn line is released (transition from the low to the high level)
- When a high level is detected on the SCLn line, the setup time for the restart condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The level on the SDAn line falls (from the high level to the low level)
- The hold time for the restart condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The level on the SCLn line falls (from the high level to the low level), the ICR.IICRSTAREQ bit is set to 0, and a restart-condition generated interrupt is output

Writing 1 to the ICR.IICSTPREQ bit causes the generation of a stop condition. The generation of a stop condition proceeds through the following operations:

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept at the low level
- The period at low level for the SCLn line is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The SCLn line is released (transition from the low to the high level)
- When a high level is detected on the SCLn line, the setup time for the stop condition is set as half of a bit period at the bit rate determined by the CCR2.BRR setting
- The SDAn line is released (transition from the low to the high level), the ICR.IICSTPREQ bit is set to 0, and a stop-condition generated interrupt is output

Figure 30.88 shows the timing of operations in the generation of start, restart, and stop conditions.

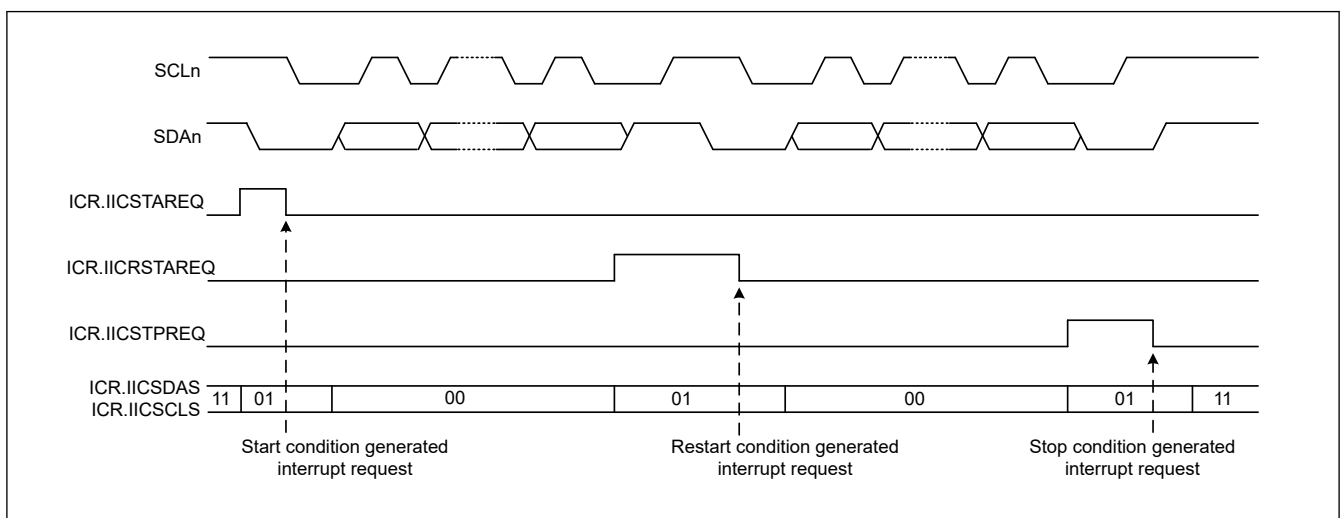


Figure 30.88 Timing of operations in generation of start, restart, and stop conditions



### 30.8.2 Clock Synchronization

The SCLn line can be driven low if a wait is inserted by a slave device at the other side of the transfer. Setting the ICR.IICCSC bit to 1 applies control to obtain synchronization when a difference arises between the levels of the internal SCLn clock signal and the level being input on the SCLn pin.

When the ICR.IICCSC bit is set to 1, the level of the internal SCLn clock signal changes from low to high. Counting to determine the period at a high level stops while the low level is being input on the SCLn pin. Counting to determine the period at a high level starts after the transition of the input on the SCLn pin to the high level.

The interval from this time until counting to determine the period at high level starts on the transition of the SCLn pin to the high level, is the total time which contains the SCLn input delay, delay for noise filtering of the input on the SCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SCLn clock is extended even when other devices do not place the low level on the SCLn line.

If the ICR.IICCSC bit is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SCLn pin and the internal SCLn clock. If the ICR.IICCSC bit is 0, synchronization with the internal SCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a wait period into the interval until the transition of the internal SCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a wait period after the transition of the internal SCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the waiting period, generation of the condition itself is not guaranteed.

Figure 30.89 shows an example operation for synchronizing the clocks.

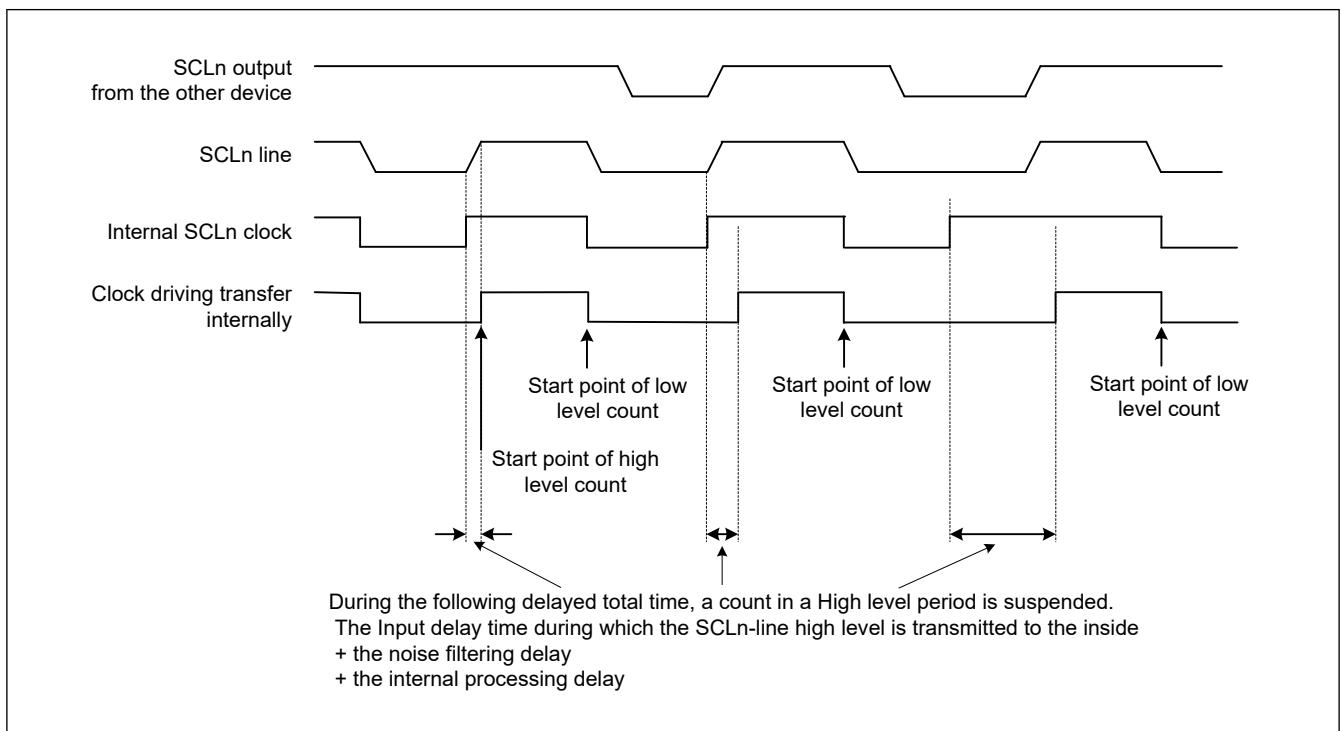


Figure 30.89 Example operations for clock synchronization

### 30.8.3 SDAn Output Delay

The ICR.IICDL[4:0] bits can be used to set a delay for output on the SDAn pin relative to falling edges of output on the SCLn pin. Delay settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, TCLK, by the divisor selected in the CCR2.CKS[1:0] bits). A delay for output on the SDAn pin applies to the start condition/restart condition/stop condition signal, 8-bit transmit data, and acknowledge bit.

If the SDAn output delay is shorter than the time for the level on the SCLn pin to fall, the change of the output on the SDAn pin starts while the output level on the SCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SDAn pin specify times greater than the time output on the SCLn pin takes to fall (300 ns for IIC in normal mode and fast mode).

Figure 30.90 shows the timing of delays in SDAn output.

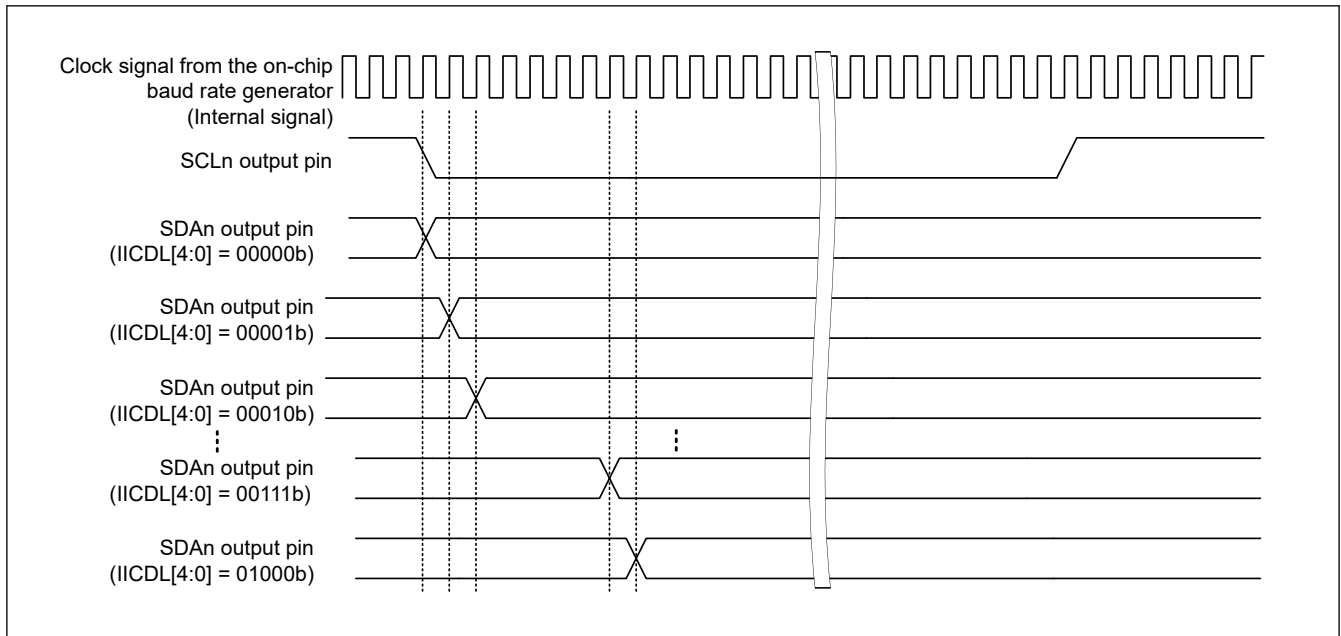


Figure 30.90 Timing of delays in SDAn output

### 30.8.4 SCI Initialization in Simple IIC Mode

Before transferring data, write the initial value 0x00 to CCR0 and initialize the interface following the example shown in Table 30.41.

Before making any changes to the operating mode or transfer format, be sure to set CCR0 to its initial value. In simple IIC mode, the open-drain setting for the communication ports should be made on the port side.

Table 30.41 Example flow of SCI initialization in simple IIC mode

No.	Step Name	Description
1	Start initialization	—
2	Set CCR0	Set CCR0.TEIE, TIE, RIE, TE, RE to 0. If you have not changed from the initial settings, you can skip this step.
3	Set ICR	Set the IICSDAS[1:0] and IICSCLS[1:0] to 11b. Set the IICDL[4:0] and IICINTM as required. Set the IICACKT and the IICCSC bits to 1.
4	Set CCR3	Set the transmission / reception format as the communication mode (MOD [2:0] = 100b) and CKE [1:0] = 00b.
5	Set CCR2	Set the bit rate modulation function*1, the clock selection, and the bit rate.
6	Set CCR1	Set noise filter, communication pin status, parity check, and CTSn / RTSn function.
7	Set the I/O port functions	Set I/O port settings that allow use (on NMOS open-drain output pins and Hi-Z) of the SCLn and SDAn pin functions.
8	Set CFCLR, ICFLR	Write 1 to the following bits and clear the corresponding flag. CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC ICFLR.IICSTIFC
9	Set CCR0 (TE, RE, TIE, RIE)	Set the TE and RE bits to 1. To enable interrupts, set the TE, TIE, RE and RIE bits to 1 with one instruction at the same time (for transmission and when the IICINTM bit is 1, clear the RIE bit). Setting the TE and RE bits to 1 makes the SCLn and SDAn pins functions available.
10	Initialization completed	—

Note: Set the CCR0.TE and RE bits to 0 or 1 at the same time.

Note 1. If you do not use the bit rate modulation function, you do not need to set it.

### 30.8.5 Operation in Master Transmission in Simple IIC Mode

Figure 30.91 and Figure 30.92 show examples of master transmission and Figure 30.93 shows an example flow of data transmission.

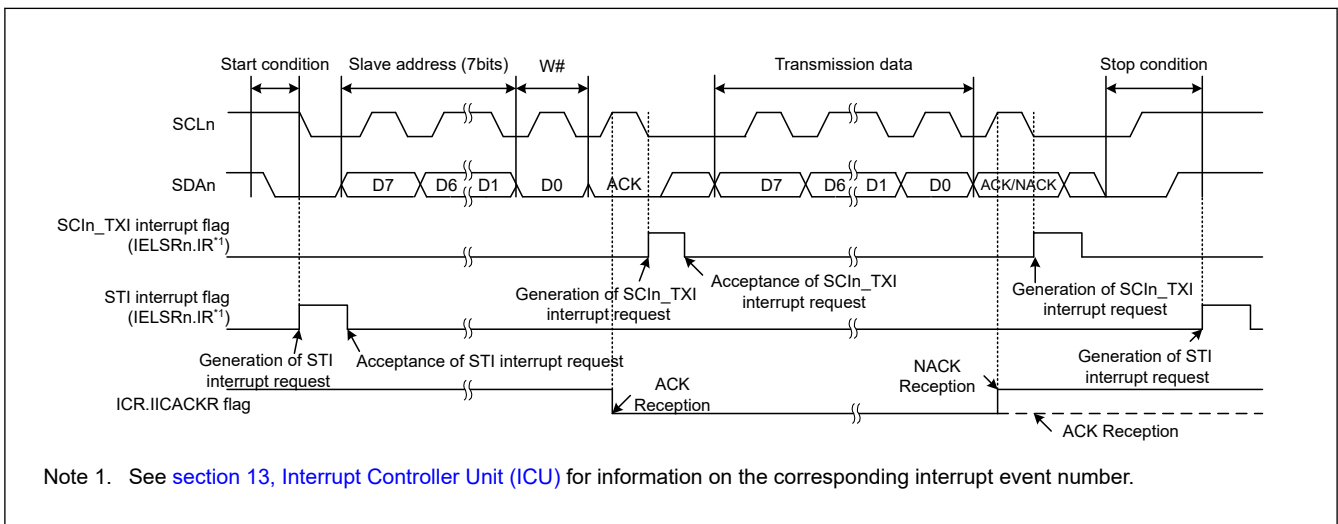
Figure 30.91 shows the operation example when ICR.IICINTM bit is 1 (use reception and transmission interrupts). In this case, you can start DMAC or DTC by SCIn\_TXI interrupt. However, if use DMAC or DTC, ACK/NACK cannot be confirmed. If you want to confirm ACK/NACK, prepare the transmit data by CPU.

In simple IIC mode, SCIn\_TXI interrupt is generated when communication of one frame is completed. And it is not used SCIn\_RXI interrupt in master transmission, so the CCR0.RIE set to 0.

See Table 30.46 for more information on the STI interrupt.

Figure 30.93 shows a flow chart in the case of ICR.IICINTM is 1 and address transmission by CPU and data transmission by DTC or DMAC. When 10-bit slave addresses are in use, steps [3] and [4] are repeated twice.

In simple IIC mode, the transmit data empty interrupt (SCIn\_TXI) is generated when communication of one frame is complete, unlike the SCIn\_TXI interrupt request generation timing during clock synchronous transmission.

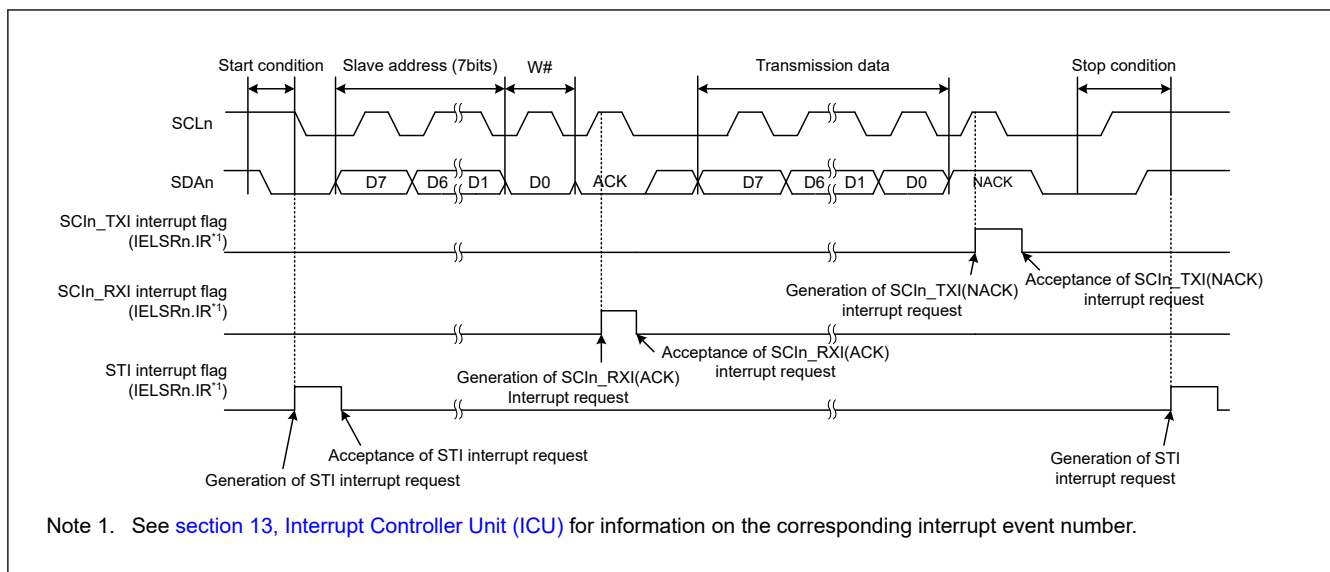


**Figure 30.91 Example 1 of operations for master transmission in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts (ICR.IICINTM = 1)**

When the ICR.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC or DMAC is activated by the ACK interrupt as the trigger and required number of data bytes are transmitted. When the NACK is received, error processing such as transmission stop and retransmission is performed using the NACK interrupt as the trigger.

To restart communication for some reason after writing data in the TDR register, use the following procedure:

1. Set the TE and RE bits in the CCR0 register to 0 to stop communication.
2. Set ICR.IICSCLS[1:0] and ICR.IICSDAS[1:0] bits to 11b, release the I<sup>2</sup>C bus, and clear the generation of a condition.
3. If the RDRF flag in the CSR register is set to 1, the RDR register is read by dummy and the RDRF bit is set to 0.
4. Set the TE and RE bits in the CCR0 register to 1 and start the next communication.



**Figure 30.92 Example 2 of operations for master transmission in simple IIC mode with 7-bit slave addresses, ACK interrupts, and NACK interrupts (ICR.IICINTM = 0)**

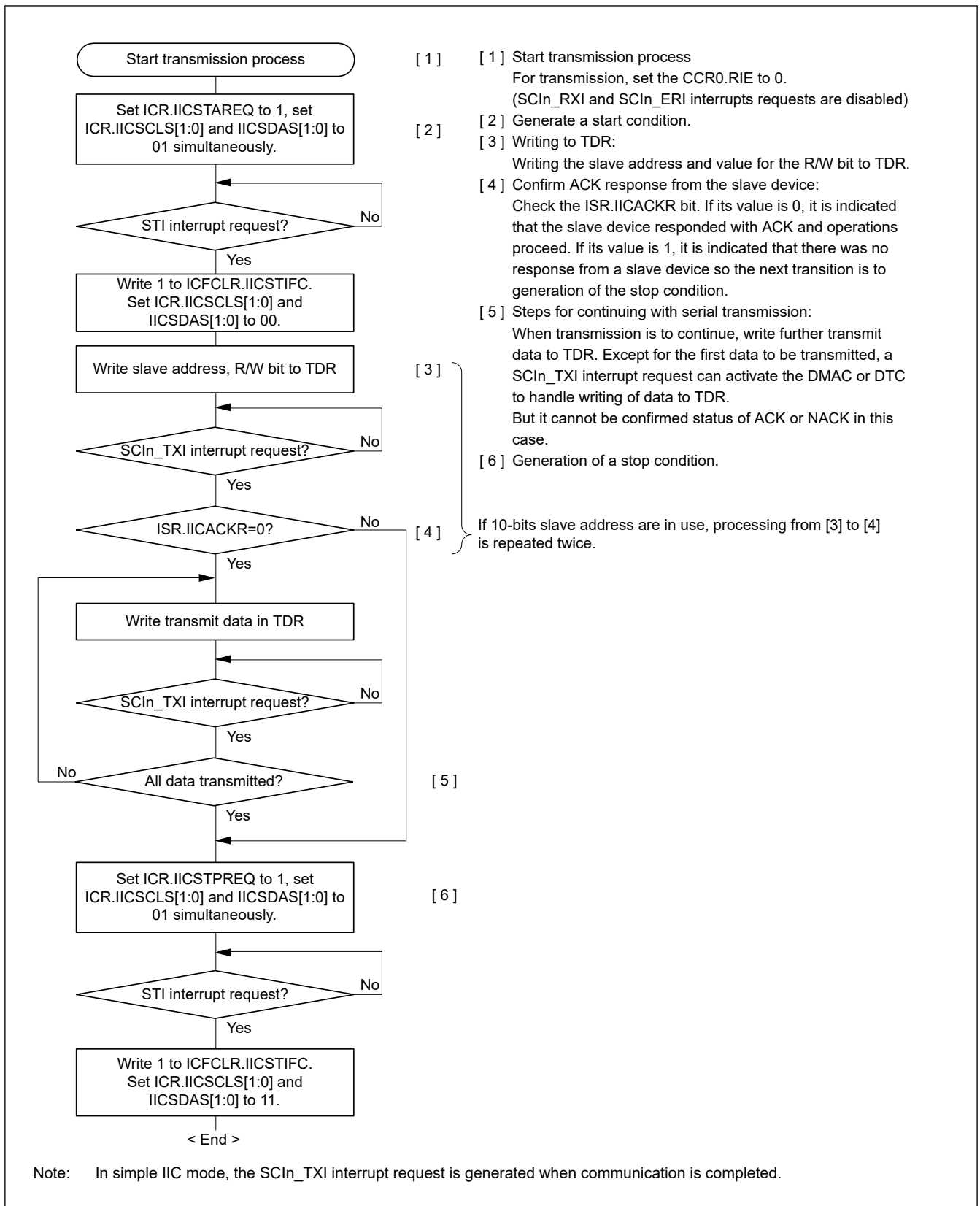


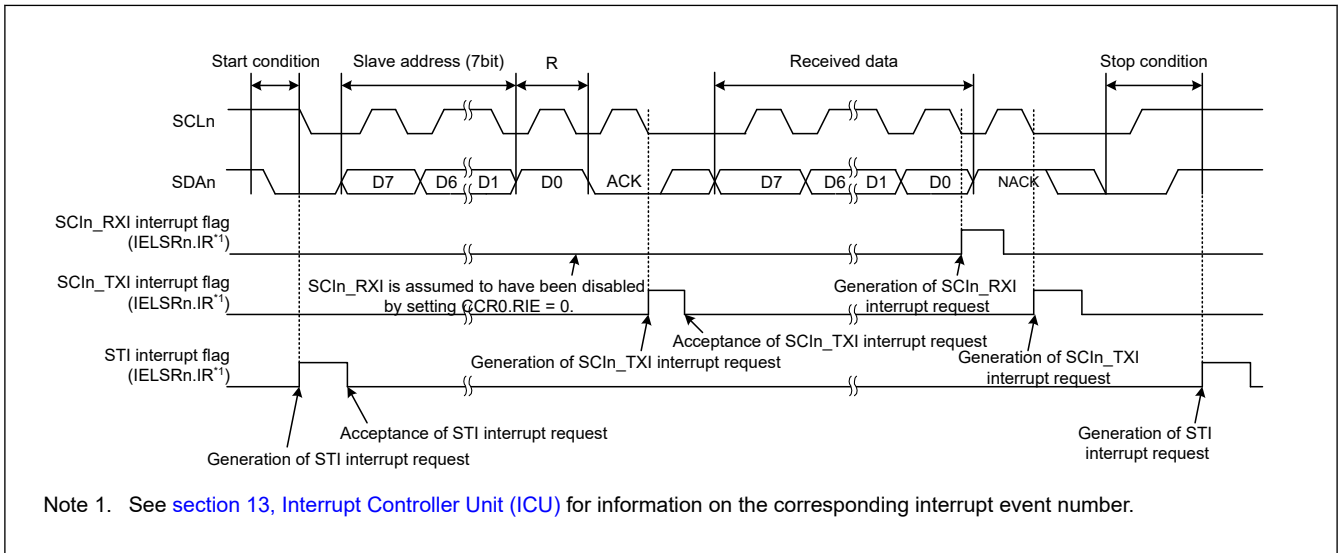
Figure 30.93 Example flow of master transmission in simple IIC mode with transmission interrupts and reception interrupts

### 30.8.6 Master Reception in Simple IIC Mode

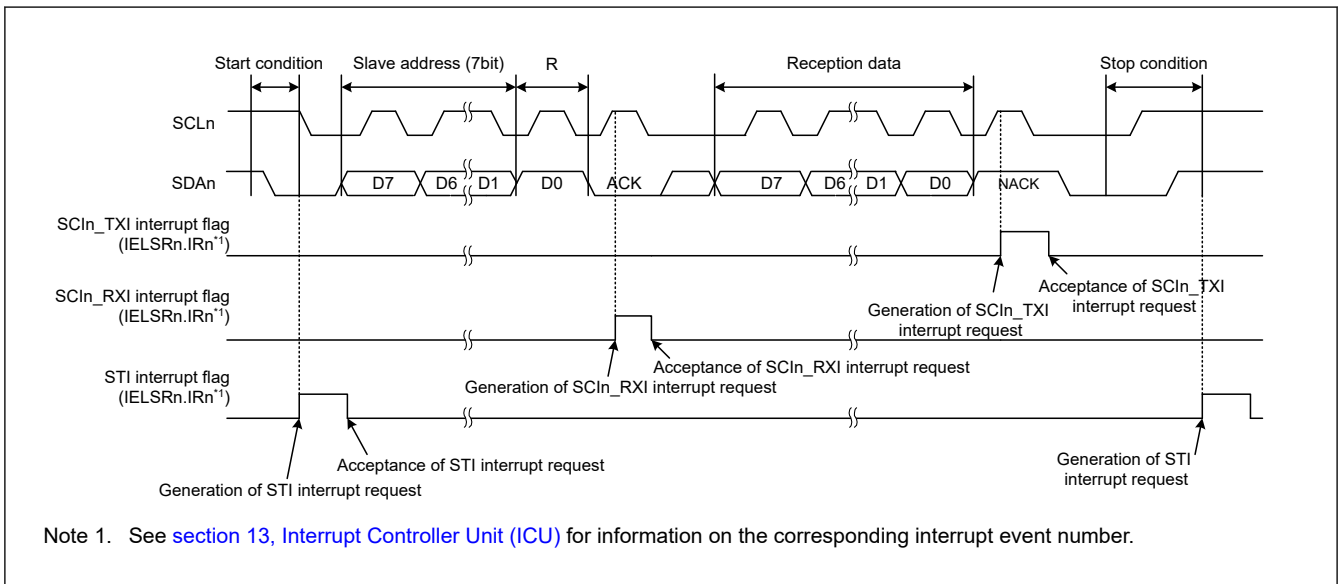
Figure 30.94 shows an example operation in simple IIC mode master reception and Figure 30.96 shows an example flow of master reception.

The value of the ICR.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and 0 (use ACK and NACK interrupts).

In simple IIC mode, the transmit data empty interrupt (SCIn\_TXI) is generated when communication of one frame is complete, unlike the SCIn\_TXI interrupt request generation timing during clock synchronous transmission.



**Figure 30.94 Example operations for master reception in simple IIC mode with 7-bit slave addresses, transmission interrupts, and reception interrupts (ICR.IICINTM = 1)**



**Figure 30.95 Example of Operations for Master Reception in Simple IIC Mode (7bit Slave address, ACK and NACK interrupt in use (ICR.IICINTM = 0))**

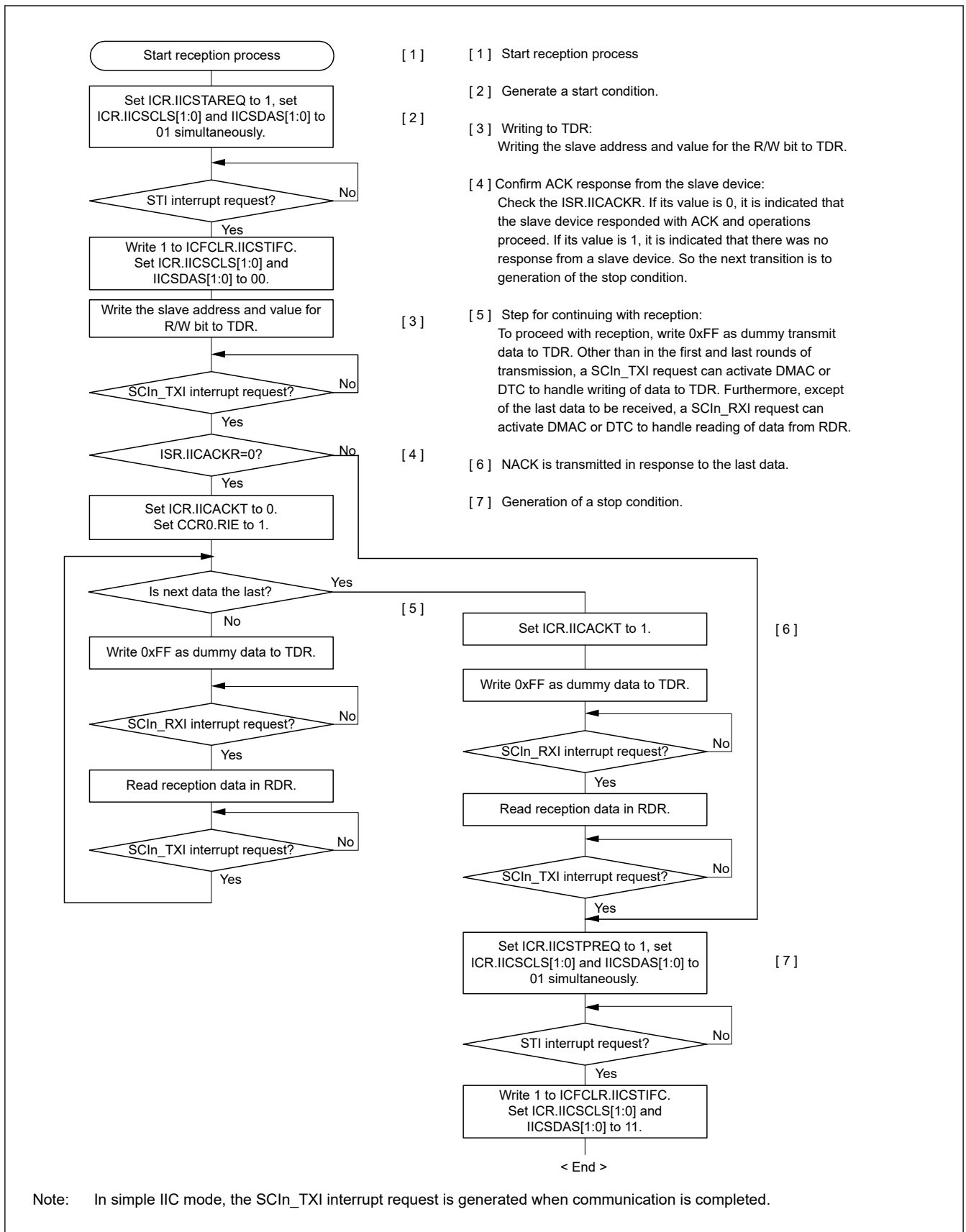
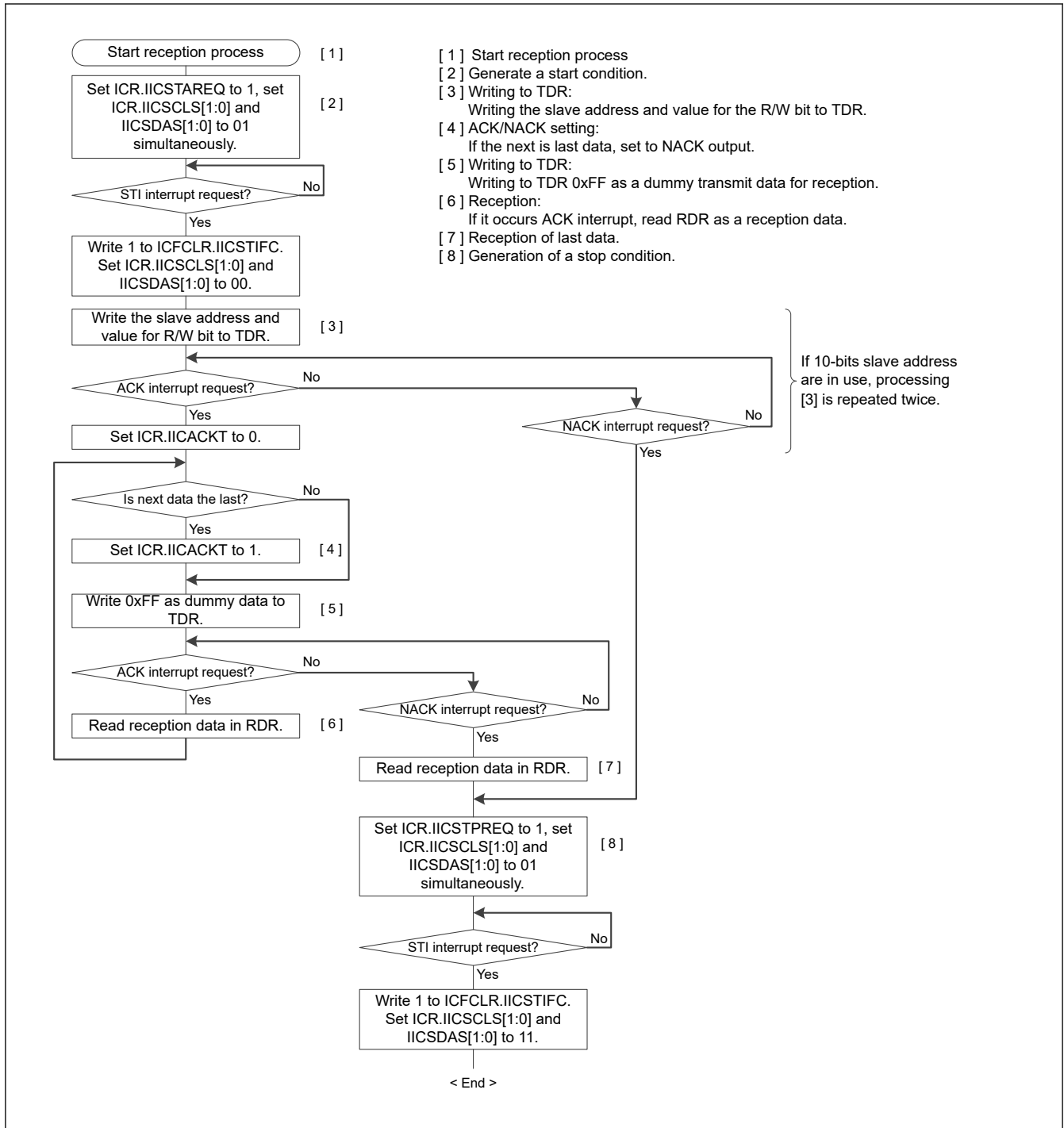


Figure 30.96 Example flow of master reception in simple IIC mode with transmission interrupts and reception interrupts (ICR.IICINTM = 1)



**Figure 30.97 Example flow of master reception in simple IIC mode with ACK Interrupts and NACK Interrupts (ICR.IICINTM = 0)**

### 30.9 Operation in Simple SPI Mode

As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Using the settings for Simple SPI mode setting (CCR3.MOD[2:0] bit = 011b) and setting the CCR0.SSE bit to 1 place the SCI in simple SPI mode. However, the SS<sub>n</sub> pin function on the master side is not required for connection of the device used as the master in simple SPI mode when the configuration only has a single master. Therefore, set the CCR0.SSE bit to 0 in such cases.

Figure 30.98 shows an example of connections for simple SPI mode. Control a general port pin to produce the SS<sub>n</sub> output signal from the master.



In simple SPI mode, data is transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended. The data can be inverted by setting the CCR3.SINV bit to 1.

Because the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a shared clock signal. Additionally, because both the transmitter and receiver have a buffered structure, writing the next transmit data while transmission is in progress and reading previously received data while reception is in progress are both possible. This enables continuous transfer.

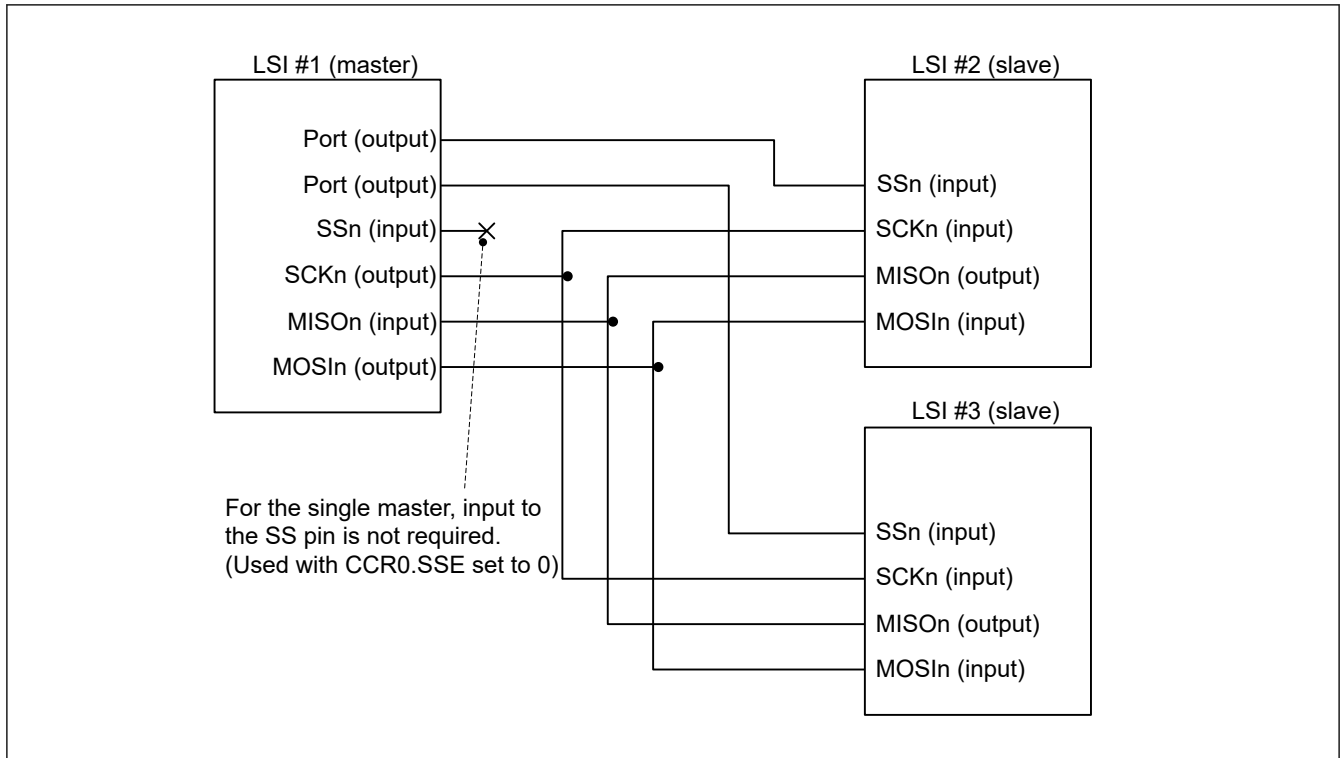


Figure 30.98 Example connections using simple SPI mode in single master mode with CCR0.SSE bit = 0

### 30.9.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (CCR3.CKE[1:0] = 00b or 01b) or slave (CCR3.CKE[1:0] = 10b or 11b).

Table 30.42 lists the relationship between the pin states, mode, and level on the SSn pin.

Table 30.42 States of pins by mode and input level on SSn pin

Mode	Input on SSn pin	State of MOSIn pin	State of MISOIn pin	State of SCKn pin
Master mode*1	High level (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (CCR0.SSE = 0), transfer is possible regardless of the input level on the SSn pin. This is equivalent to input of a high level on the SSn pin. The SSn pin is not used and is available for other purposes.

Note 2. The MOSIn pin output is in the high-impedance state when serial transmission is disabled (CCR0.TE bit = 0).

Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (CCR0.TE = 0 and CCR0.RE = 0) in a multi-master configuration (CCR0.SSE = 1).

### 30.9.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the CCR3 to 00b or 01b selects master mode operation. The SSn pin is not used in single-master configurations (CCR0.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn pin.

When the level on the SSn pin is high in a multi-master configuration (CCR0.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission.

When the level on the SSn pin is low in a multi-master configuration (CCR0.SSE = 1), there are other masters, and a transmission or reception is in progress. The MOSIn output and SCKn pins are placed in the high-impedance state and starting transmission or reception is not possible. In addition, the value of the CSR.MFF bit is 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading CSR.MFF flag. If a mode fault error occurs while transmission or reception is in progress, transmission or reception does not stop, but the MOSIn and SCKn outputs are in the high-impedance state after completion of the transfer. When the SSn pin input becomes high level, the SCKn pin outputs a clock signal and the MOSIn outputs data. Even if the SCKn pin and the MOSIn pin are in the high impedance state, internal transmission or reception operation continues, but it stops after transmission or reception of a single character is complete. In this case, any of SCIn\_TXI, SCIn\_RXI, and SCIn\_TEI interrupts occurs.

Use a general port pin to produce the SS output signal from the master.

### 30.9.3 SS Function in Slave Mode

Setting the CCR3.CKE[1:0] bits to 10b or 11b selects slave operation. When the SSn pin is high, the MISO output pin is in the high-impedance state and clock input through the SCKn pin is ignored. When the SSn pin is low, clock input through the SCKn pin is valid and transmission or reception can proceed.

If the input on the SSn pin changes from low to high during transmission or reception, the MISO output pin is placed in the high-impedance state. Transmission / reception operation is immediately suspended. If the transmission is in progress, the CSR.TEND flag will not be set, a transmit end interrupt will not be output, and an abnormal stop status will occur. So, do not negate the SSn pin during slave transmission / reception. If an abnormal stop occurs, set CCR0.RE and CCR0.TE to 0 to stop transmission / reception. To resume transmission / reception, set CCR0.RE and CCR0.TE to 1 after at least  $TCLK \times 3$  cycles +  $PCLK \times 3$  cycles.

### 30.9.4 Relationship between Clock and Transmit/Receive Data

The CPOL and CPHA bits in the CCR3 register can be used to set up the clock for use in transmission and reception in four different ways. The relation between the clock signal and the transmission and reception of data is shown in [Figure 30.99](#).

The relation is the same for both master and slave operation. This is the same as when the level on the SSn pin is high.

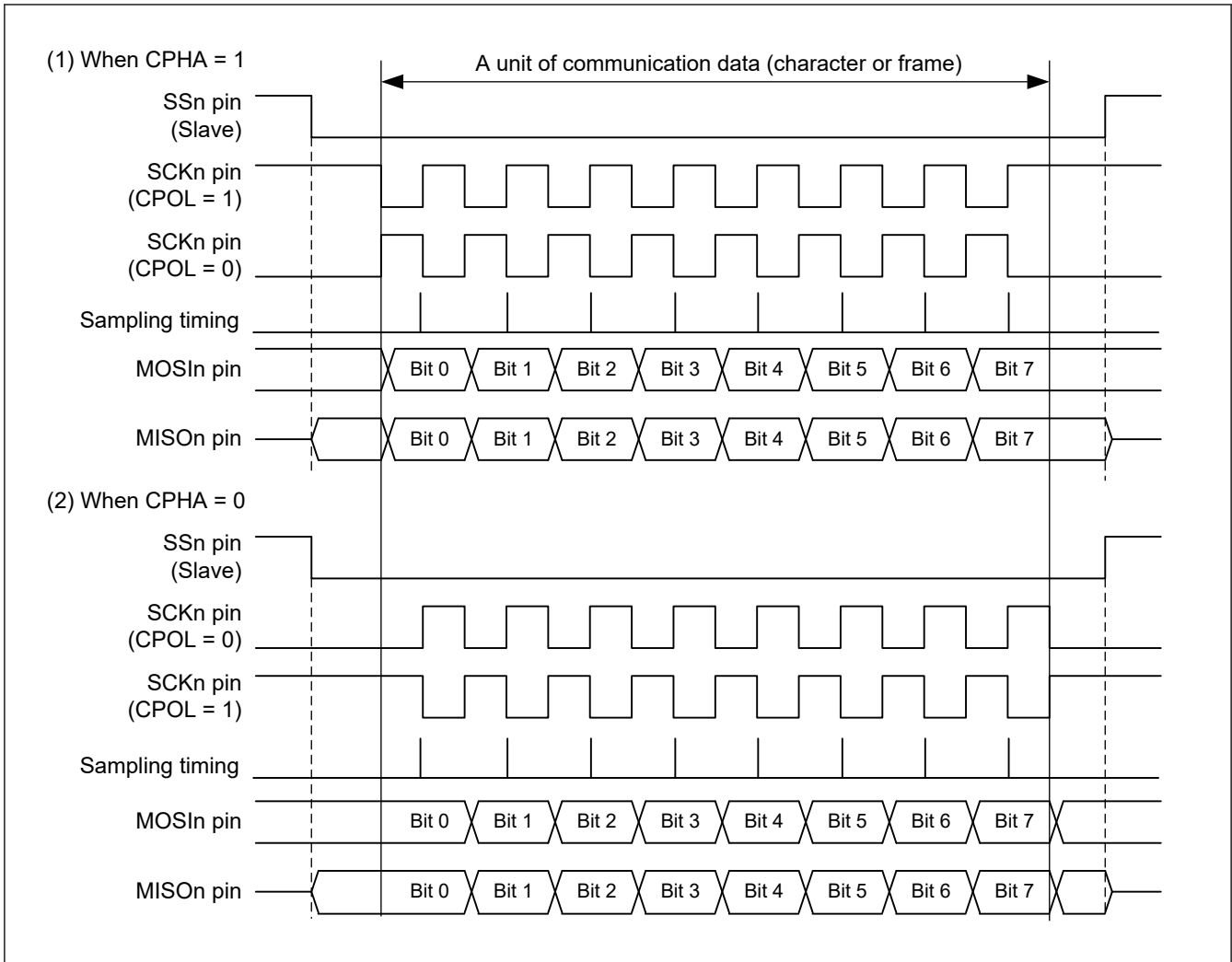


Figure 30.99 Relation between clock signal and transmit or receive data in simple SPI mode

### 30.9.5 SCI Initialization in Simple SPI Mode

Initialization in simple SPI mode is the same as in clock synchronous mode. See [section 30.6.3. SCI Initialization in Clock Synchronous Mode](#) for an example initialization flow. The CPOL and CPHA bits in the CCR3 register must be set to ensure that the clock signal is suitable for both master and slave devices.

Always initialize the CCR0 register before making any changes to the operating mode or transfer format.

Note: Only the RE bit is set to 0. The CSR.ORER, FER, PER, and RDR flags are not initialized.

Changing the value of the TE bit from 1 to 0 or from 0 to 1 when the TIE bit in the CCR0 register is 1 at the same time, leads to the generation of a transmit data empty interrupt (SCIn\_TXI).

### 30.9.6 Transmission and Reception of Serial Data in Simple SPI Mode

In master operation, ensure that the SSn pin of the slave device on the other side of the transfer is at the low level before starting the transfer and at the high level on completion of the transfer. In multiple master operation with CCR0.SSE = 1 even in master mode, a mode fault error will occur if the SSn pin goes low. Therefore, make sure that no mode fault error has occurred before starting communication, and start communication, and make sure that no mode fault error has occurred even after communication ends. If a mode fault error has occurred, communication may be incomplete, so measures such as retransmission are required. Otherwise, the procedures are the same as in clock synchronous mode.

In slave mode, it operates according to the SSn pin input level. Other steps are the same as those of clock synchronous mode.

### 30.9.7 Reception Sampling Timing Adjustment Function in Simple SPI Mode with internal clock used

The reception sampling timing adjustment function in simple SPI mode is the same as the reception sampling timing adjustment function in clock synchronous mode. For the description of operation, see [section 30.6.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with internal clock used.](#)

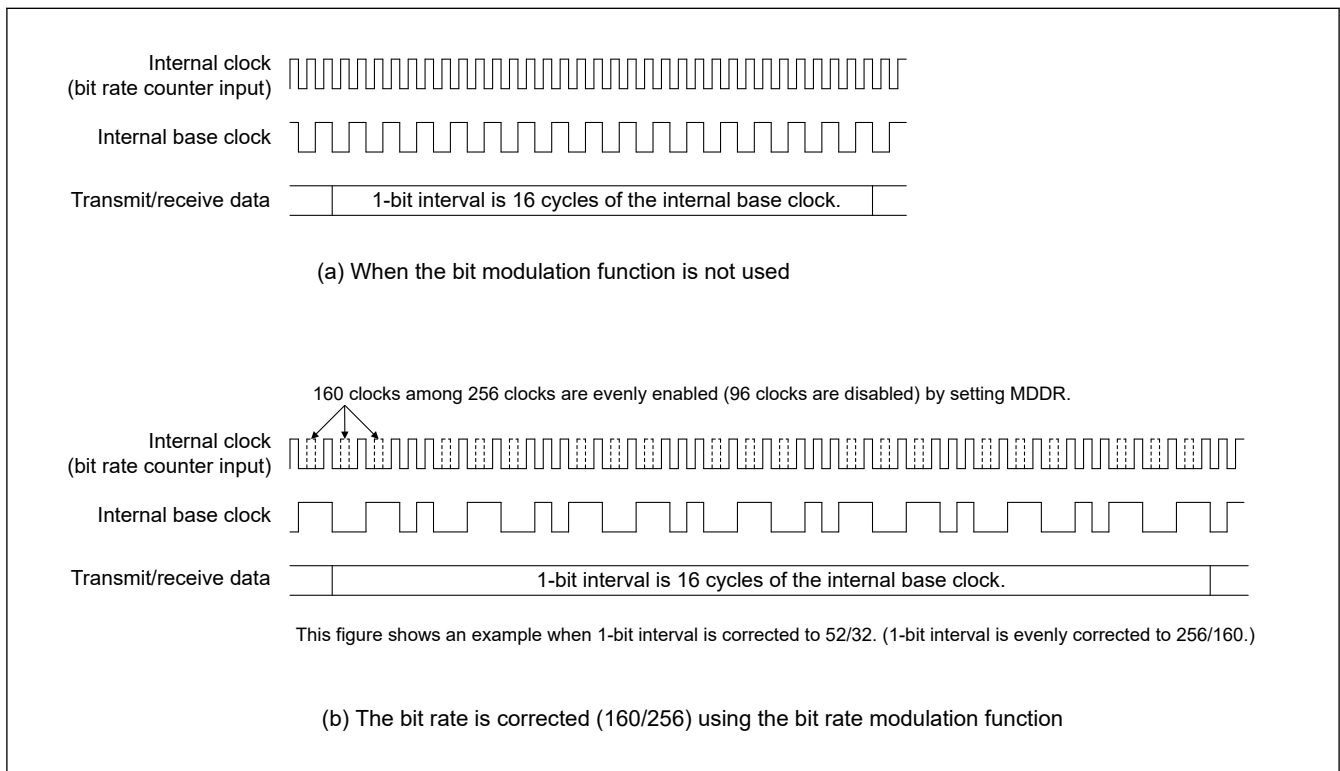
### 30.10 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be evenly corrected using the number specified in the MDDR register among 256 clock cycles of internal clocks which is selected by the CKS[1:0] bits in CCR2.

[Figure 30.100](#) shows an example where the PCLK is selected in the CKS[1:0] bits in CCR2, the BRR bit is set to 0, and the MDDR is set to 160 in Asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256).

**Note:** Enabling an internal clock causes bias, and expansion and contraction are generated in the pulse width of the internal base clock.

Do not use this function in clock synchronous mode, simple SPI mode, Smart Card Interface mode, Manchester mode and Simple LIN mode.



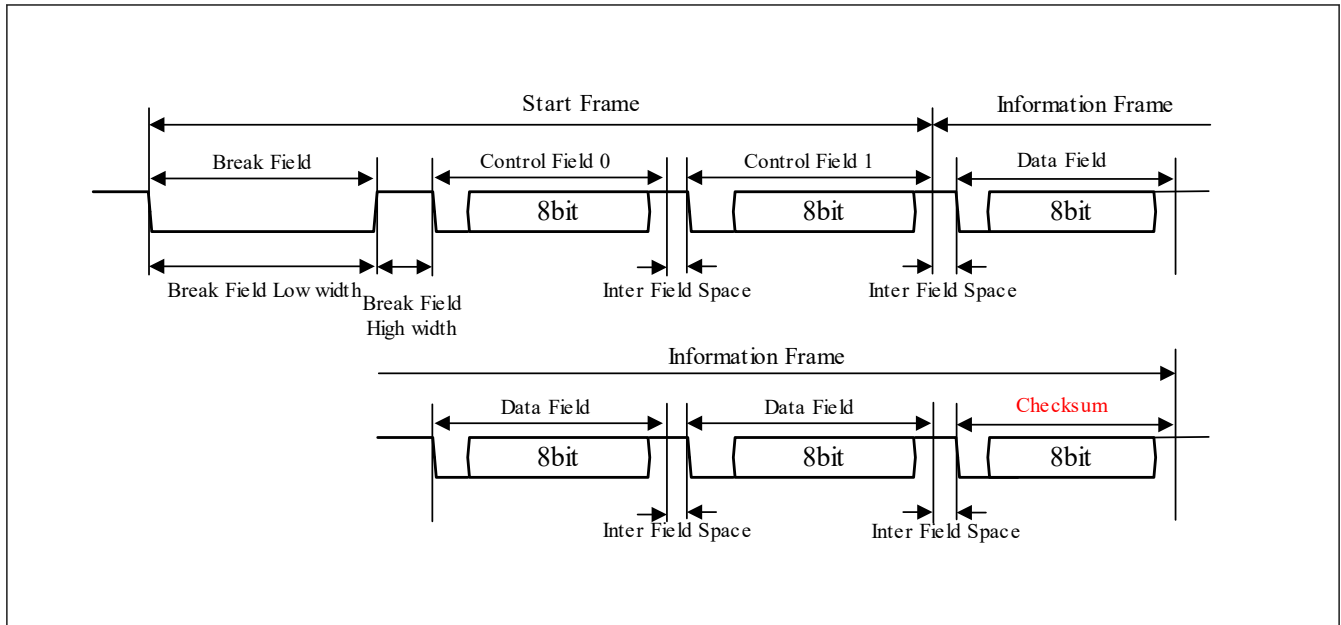
**Figure 30.100 Example internal base clock when bit rate modulation function is used**

### 30.11 Simple LIN mode

As an extended function of the SCI, the SCI supports the serial communication protocol ([Figure 30.101](#)) consisting of a Start Frame and an Information Frame as Simple LIN. Simple LIN mode is enabled by the CCR3.MOD[2:0] = 110b. Since the Simple LIN mode uses the same circuit as the Asynchronous mode for transmission / reception control other than Break Field, the basic communication settings required for the Asynchronous mode are also required for the Simple LIN mode.

(For the setting value when using simple LIN, see the explanation in [section 30.2. Register Descriptions](#). In particular, note that CCR3.RXDESEL needs to be changed from the initial value and set to 1.)

The Start Frame consists of a Break Field, Control Field 0, and Control Field 1. The Information Frame can be configured with some Data Fields, a Checksum Field.



**Figure 30.101 Simple LIN Protocol Example**

The following describes operations when the Simple LIN is used. In this section, operations are described with the following conditions:

Communication pin (RXDn / TXDn) level inversion function: OFF (RINV = TINV = 0)

When using the simple LIN with the communication pin (RXDn / TXDn) level inversion function enabled, replace the RXDn and TXDn signal levels with their inverted levels.

### 30.11.1 Simple LIN Start Frame Transmission

Figure 30.102 shows an example of transmission of the Start Frame consisting of a Break Field, Control Field0, and Control Filed1. (Omit Break Field and Control Field0 according to the Start Frame configuration.)

Figure 30.103 shows a flowchart for Start Frame transmission.

The SCI operates as follows during Start Frame transmission.

1. Make the initial settings for the SCI according to the SCI initialization flow (Figure 30.68) in Asynchronous mode. In Simple LIN mode, do not set CCR0.TE and TIE to 1 at the same time to avoid SCIn\_TXI output before the Break Field. Therefore, perform the following two steps sequentially to set the SCI initialization flow (Asynchronous mode) procedure [9].
  - Set the bits except CCR0.TIE. (CCR0.TIE = 0, CCR0.TE = 1, and CCR0.RE = 0)
  - Set CCR0.TIE to 1.
2. When 1 is written to TCST, the Break Field output timer starts counting and outputs a low level (Break Field) from the TXDn pin for the period set in XCR2.BFLW[15:0]. A timer count clock source can be selected by XCR0.TCSS[1:0]. Writing 0 to XCR1.TCST suspends output of the Break Field. After the suspension, set CCR0.TE = 0 and turn off the transmission.
3. When the Simple LIN module timer count value matches the set XCR2.BFLW[15:0] value, the timer stops counting and inverts the TXDn pin output level, and the XSR0.BFOF flag is set to 1<sup>\*1</sup>. Furthermore, if XCR0.BFOIE has been set to 1 at this time, a SCIn\_TXI interrupt is generated.
4. After the SCIn\_TXI interrupt and confirming XSR0.BFOF = 1, write the transmitted data then the Control Field 0 data is transmitted using the SCI<sup>\*2</sup>.
5. After the Control Field 0 data has been transmitted, write the Control Field 1 data to TDR. And it is transmitted.
6. After the Control Field 1 data has been transmitted, the Information Frame data is transmitted.

Note 1. After XSR0.BFOF is set to 1, if 1 is written to XCR1.TCST without clearing it, no SCIn\_TXI interrupt is output at the end of Break Field transmission. Clear XSR0.BFOF before writing 1 to XCR1.TCST.

Note 2. LIN communication requires a Break Delimiter (IDLE period) of 1 bit or more from the end of Break Field output until the next data transmission starts. For this reason, the Break Delimiter length is counted upon completion of Break Field output. If transmit data is written while the Break Delimiter length is being counted, transmission does not start until the Break Delimiter length counting is completed. When transmit data is written after the Break Delimiter length has been counted, transmission starts at the same timing as normal data transmission.

Break Delimiter length count time after Break Field output:

1-bit to 2-bit length (CCR3.STP = 0)

2-bit to 3-bit length (CCR3.STP = 1)

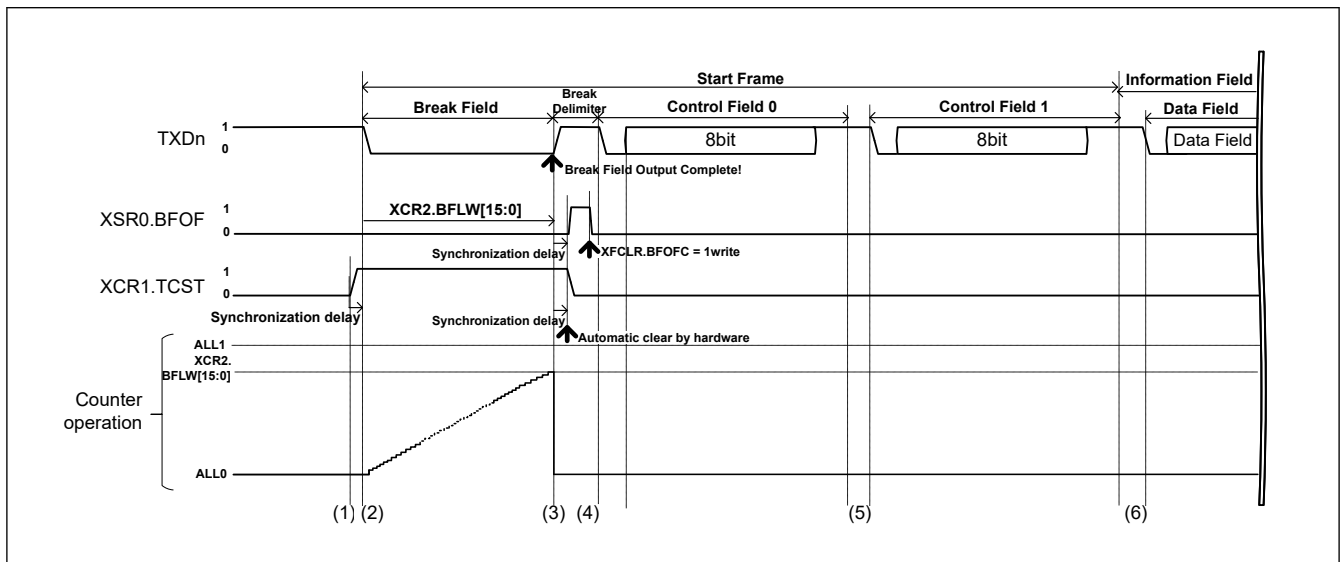


Figure 30.102 Start Frame Transmission Example

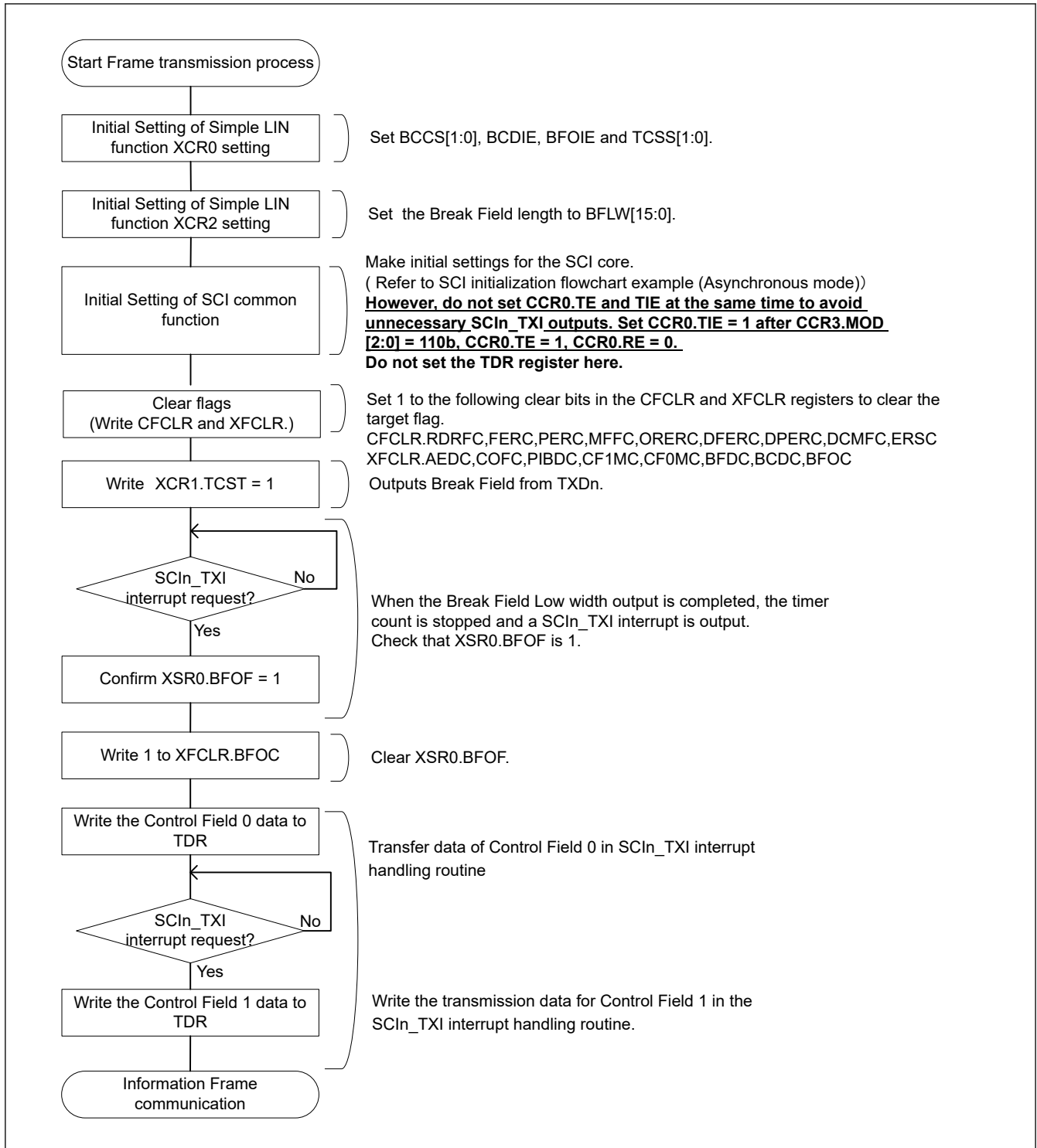
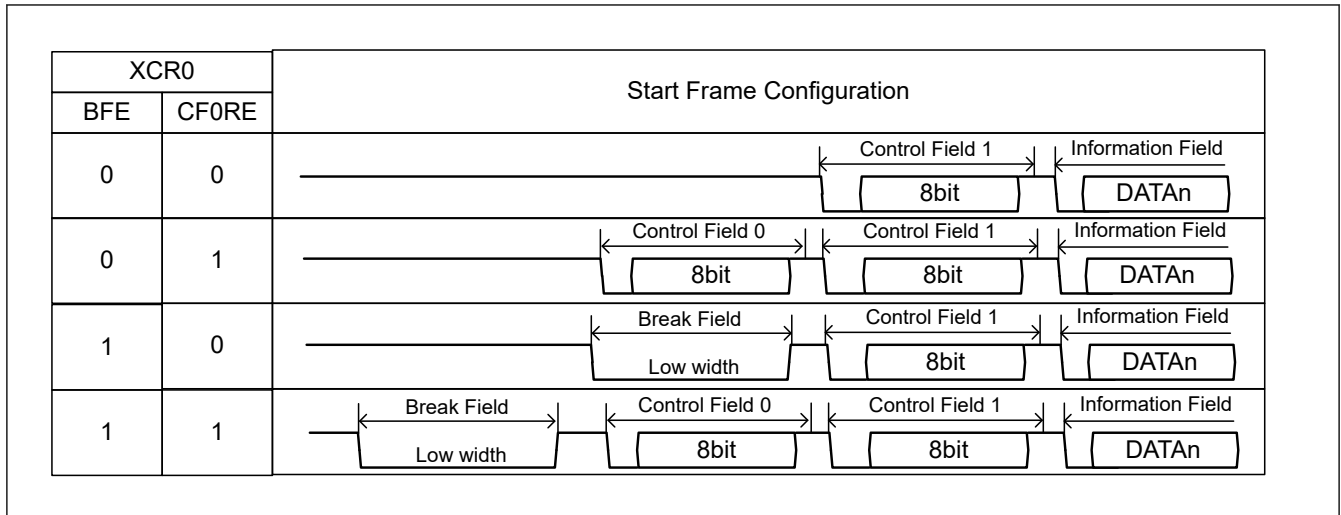


Figure 30.103 Start Frame Transmission Flowchart Example

### 30.11.2 Simple LIN Start Frame Reception

The SCI can detect Start Frames configured as shown in [Figure 30.104](#).



**Figure 30.104 Start Frame Configuration**

(1) Simple LIN normal reception of Start Frame (PIB not used)

Figure 30.105 shows an example of normal reception of the Start Frame consisting of a Break Field, Control Field0, and Control Field1. Figure 30.106 shows an example of reception to detect the Break Field during Control Field 1. Figure 30.107 shows a flowchart to receive the Start Frame, and Figure 30.108 shows a state transition diagram.

When receiving the Start Frame, the SCI operates as follows. Omit the processing of Break Field and Control Field0 according to the Start Frame configuration.

- Writing 1 to XCR1.SDST makes it possible to detect the Start Frame. When XCR0.BFE = 1, RXDn input to the SCI core is disabled until the Break Field is detected (because XSR0.RXDSF is set to 1). Once the Break Field is detected, RXDn input can be received to the SCI core (XSR0.RXDSF = 0).
- When a low level is input from the RXDn pin, the Break Field detection count starts. A timer count clock source can be selected by XCR0.TCSS[1:0].
- When a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] is input from the RXDn pin, it is determined as Break Field. At this time, XSR0.BFDF is set to 1. If XCR0.BFDIE has been set to 1 at this time, a SCIn\_BFD interrupt is generated. The timer continues counting until the RXDn rising edge or counter overflow.
- After the Break Field is detected, when the input level from the RXDn pin becomes high, the count value is captured to XSR1.TCNT[15:0] when BMEN = 0. At this time, XSR0.RXDSF is cleared to 0 and the SCI core starts receiving the RXDn input.
- The SCI core starts receiving Control Field 0. Because the simple LIN continuously counts the edge interval, it determines a low level that is equal to or longer than the period set in XCR2.BFLW[15:0] as detection of the Break Field. When the Break Field is detected in the Control Field 0 phase, the SCI core waits for reception of Control Field 0 again (Figure 30.106).
- When Control Field 0 has been received, an SCIn\_RXI interrupt is generated and the Control Field 0 data is stored in XSR0.CF0RD[7:0]. When the received data matches the set XCR2.CF0D[7:0] value, XSR0.CF0MF is set to 1. If the received data differs from the set XCR2.CF0D[7:0] value, the SCI transitions to the state before the Break Field is detected.
- The SCI core starts receiving Control Field 1. When BFE = 1, the Break Field detection function is continuously enabled while SDST = 1 as in the case of Control Field 0. When the Break Field is detected in the Control Field 1 phase, the SCI core waits for reception of Control Field 0 again.
- When Control Field 1 has been received, an SCIn\_RXI interrupt is generated and the Control Field 1 data is stored in XSR0.CF1RD[7:0]. When the received data matches the set XCR1.PCF1D[7:0] value or the set XCR1.SCF1D[7:0] value, XSR0.CF1MF is set to 1. If the received Control Field 1 data matches neither the set XCR1.PCF1D[7:0] value nor the set XCR1.SCF1D[7:0] value, the SCI transitions to the state before the Break Field is detected.
- The SCI core performs Information Frame communication.
- When communication is completed, write 0 to XCR1.SDST and 0 to CCR0.RE to stop reception.



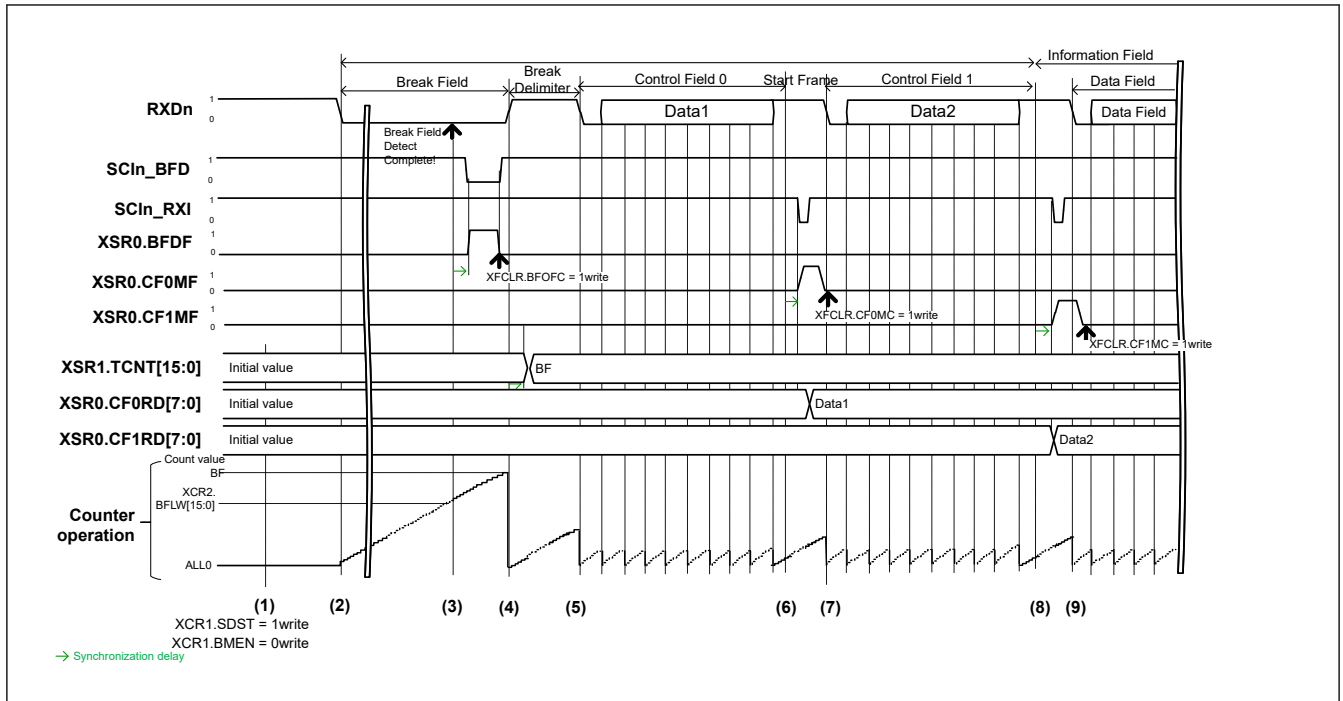


Figure 30.105 Normal Reception Example of Start Frame (PIB Not Used)

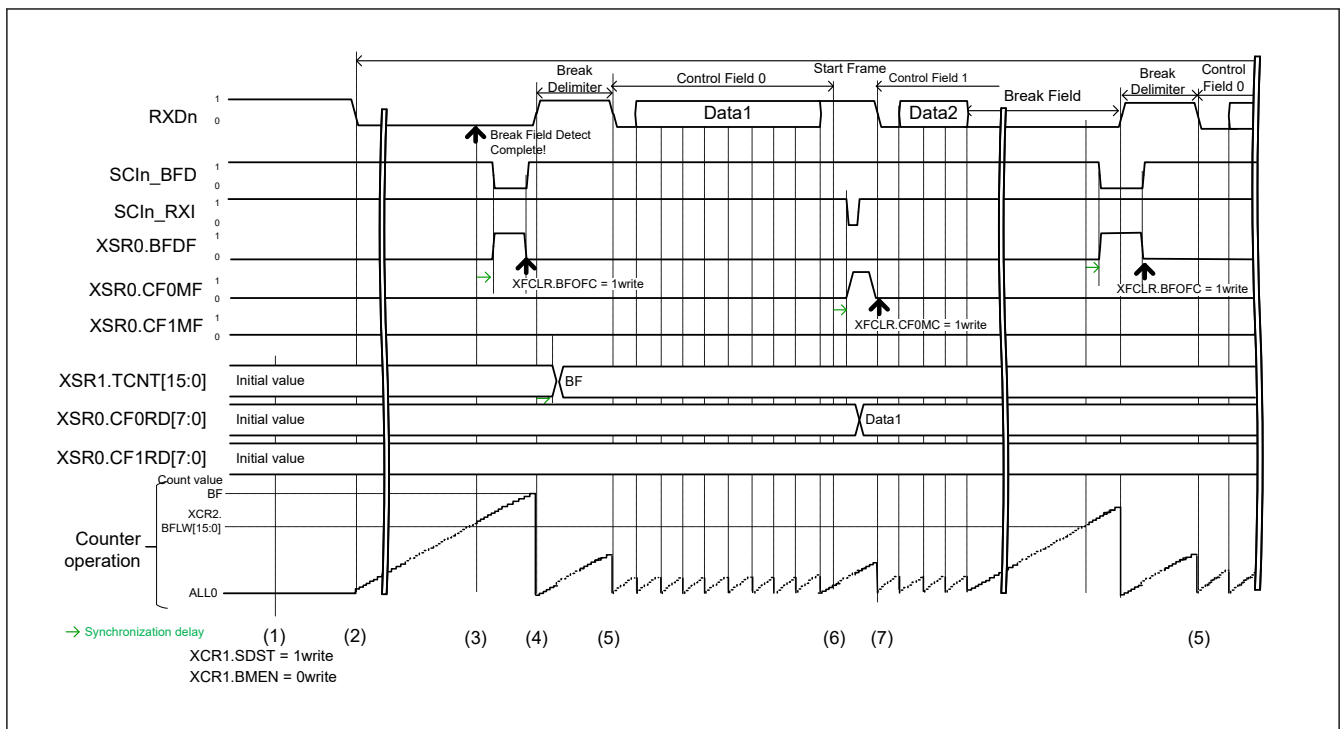


Figure 30.106 Start Frame Reception Example (PIB Not Used) Break Field Detected during Control Field 1

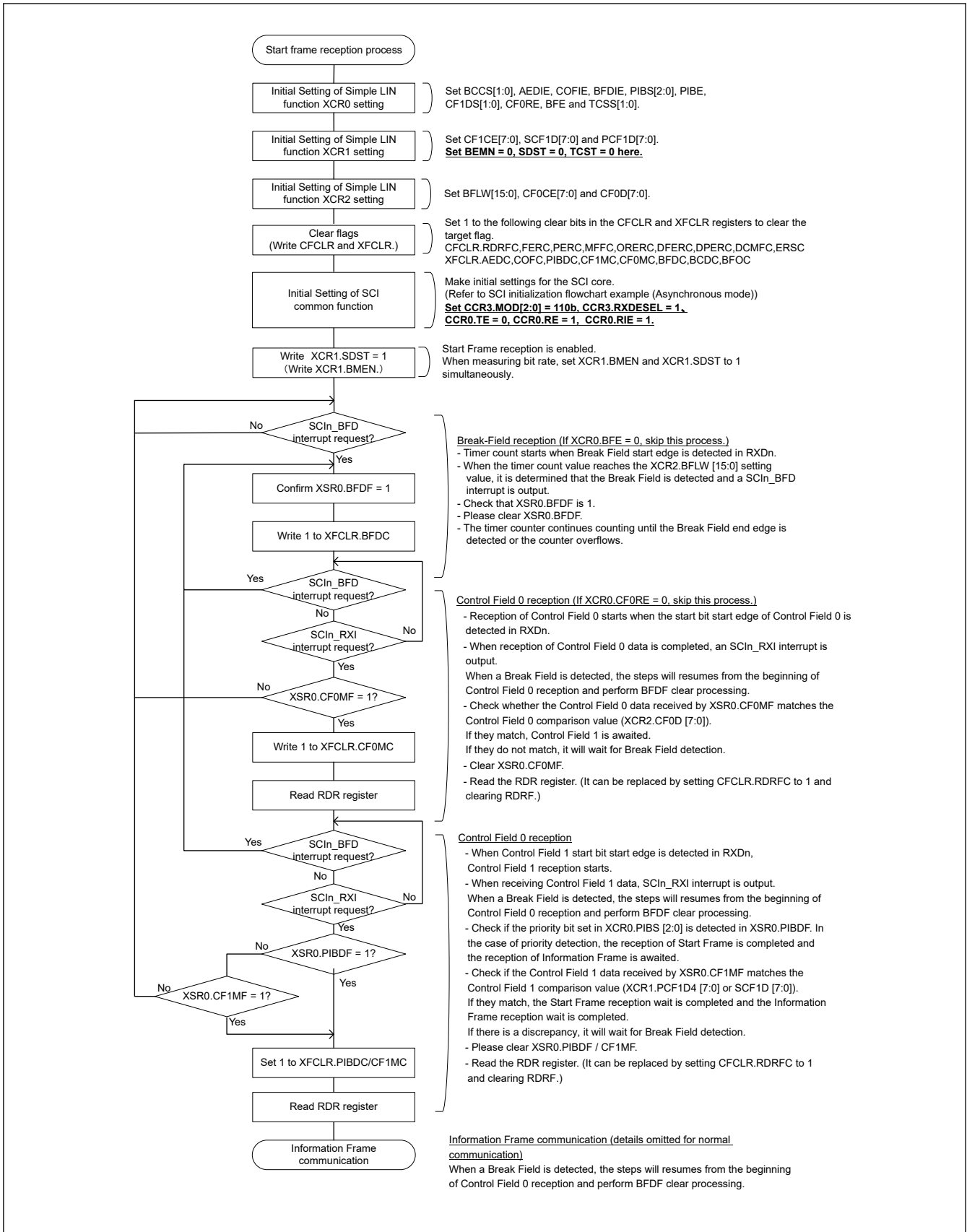
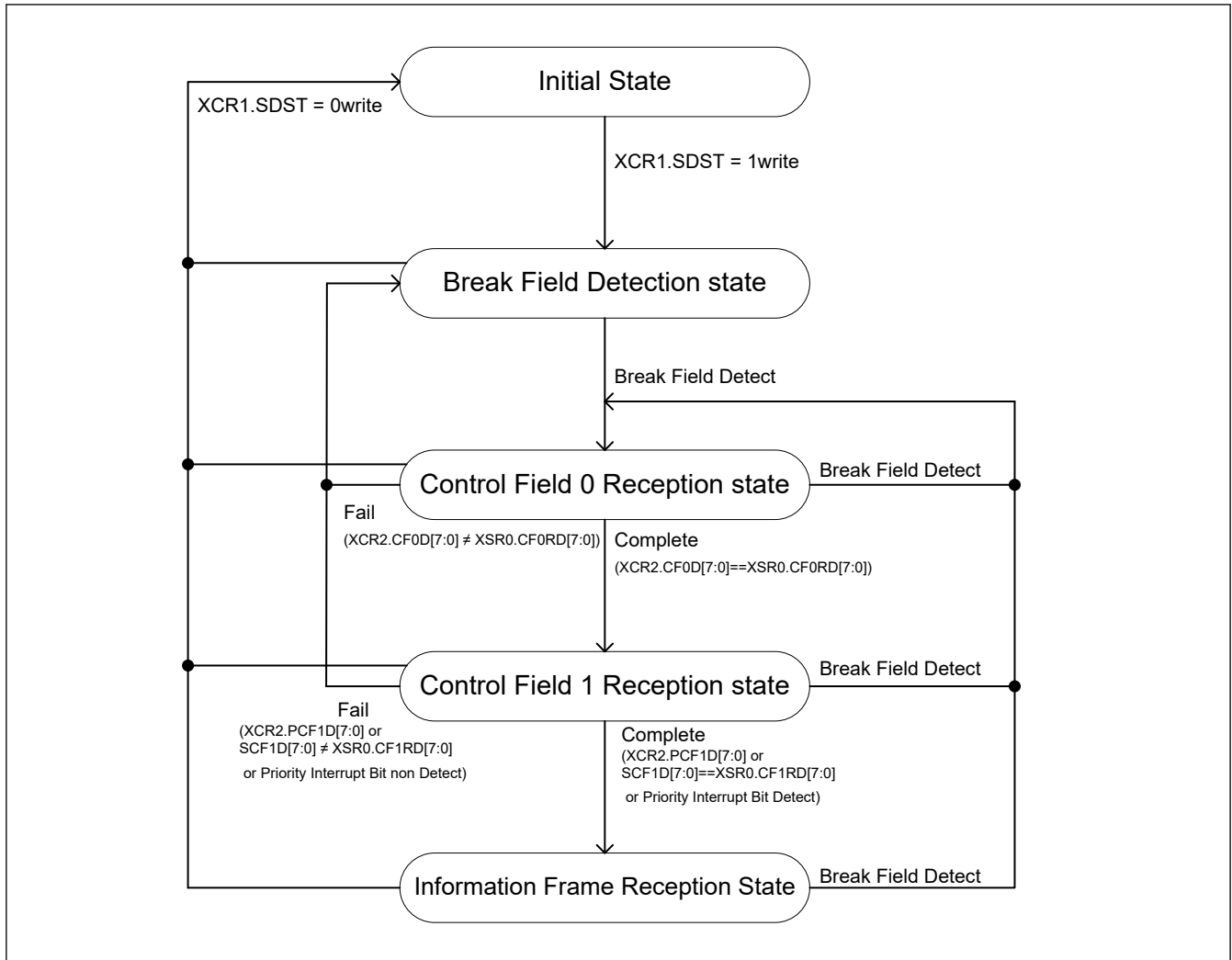


Figure 30.107 Example of Start Frame Reception Flowchart



**Figure 30.108 State Transition Diagram of Start Frame Reception**

(2) Simple LIN Start Frame reception (using the priority interrupt bit)

Figure 30.109 shows an example of Start Frame reception using the priority interrupt bit. The priority interrupt bit is enabled by setting XCR0.PIBE to 1.

The SCI operates as follows during Start Frame reception using the priority interrupt bit.

Steps (1) to (7) are the same as steps (1) to (7) in the Start Frame reception example in Figure 30.105.

(8) When the value specified in the XCR0.PIBS[2:0] bits matches the set XCR1.PCF1D[7:0] value, XSR0.PIBDF is set to 1 and the SCI core performs communication of the Information Frame. If the data received in Control Field 1 matches neither the set XCR1.PCF1D[7:0] value nor the set XCR1.SCF1D[7:0] value and the priority interrupt bit is not detected, the SCI transitions to the state before the Break Field is detected.

(9) Communicate information frame at the SCI core.

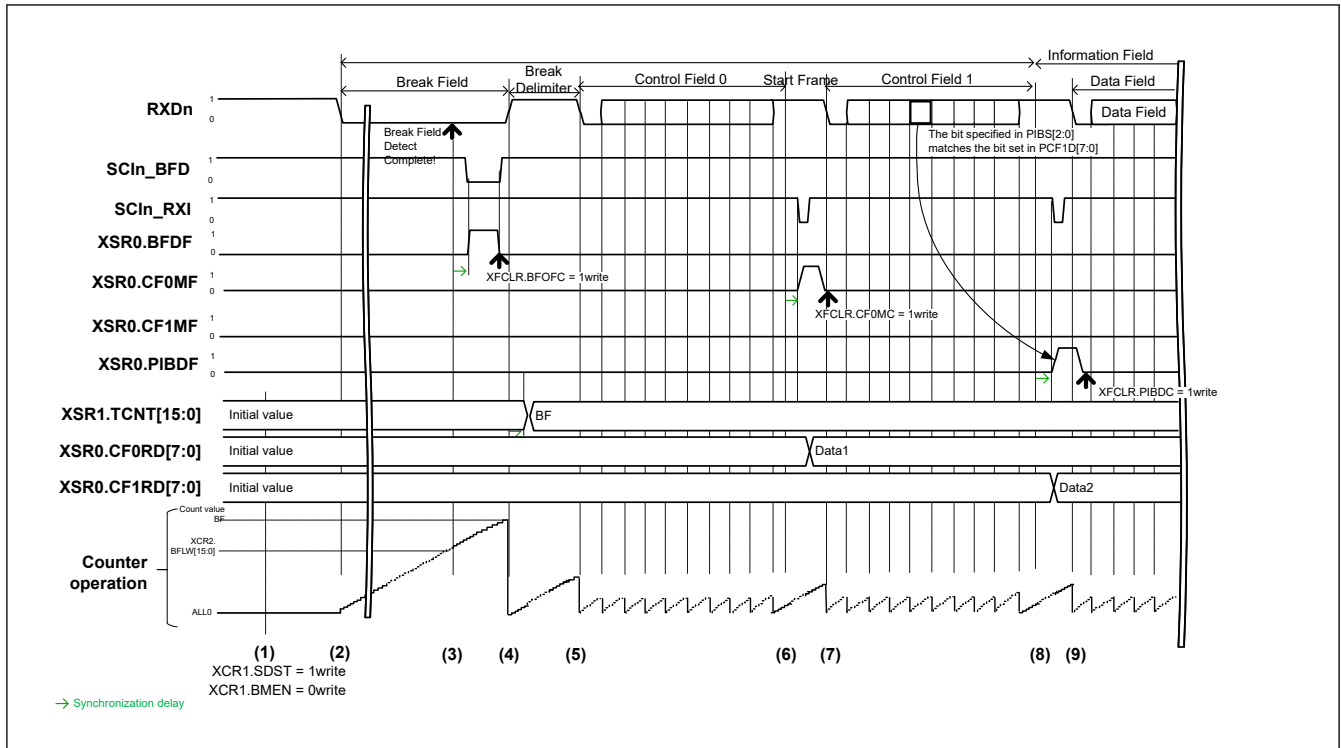


Figure 30.109 Start Frame Reception Example (Priority Interrupt Bit Used)

### 30.11.3 Simple LIN Bus Conflict Detection Function

In Simple LIN mode (CCR3MOD[2:0] = 110) when TE = 1, the bus conflict detection function works during Break Field output and during data transmission.

Figure 30.110 shows an operation example of the bus conflict detection function. The TXDn pin output and the RXDn pin input are sampled by the bus conflict detection clock set in XCR0.BCCS[1:0]. When a mismatch occurs three times in a row, XSR0.BCDF is set to 1, and if XCR0.BCDIE has been set to 1 at this time, an SCIn\_ERI interrupt is generated.

When an SCIn\_ERI interrupt is generated, stop transmission according to Figure 30.111. Check the bus state to decide whether to resume transmission.

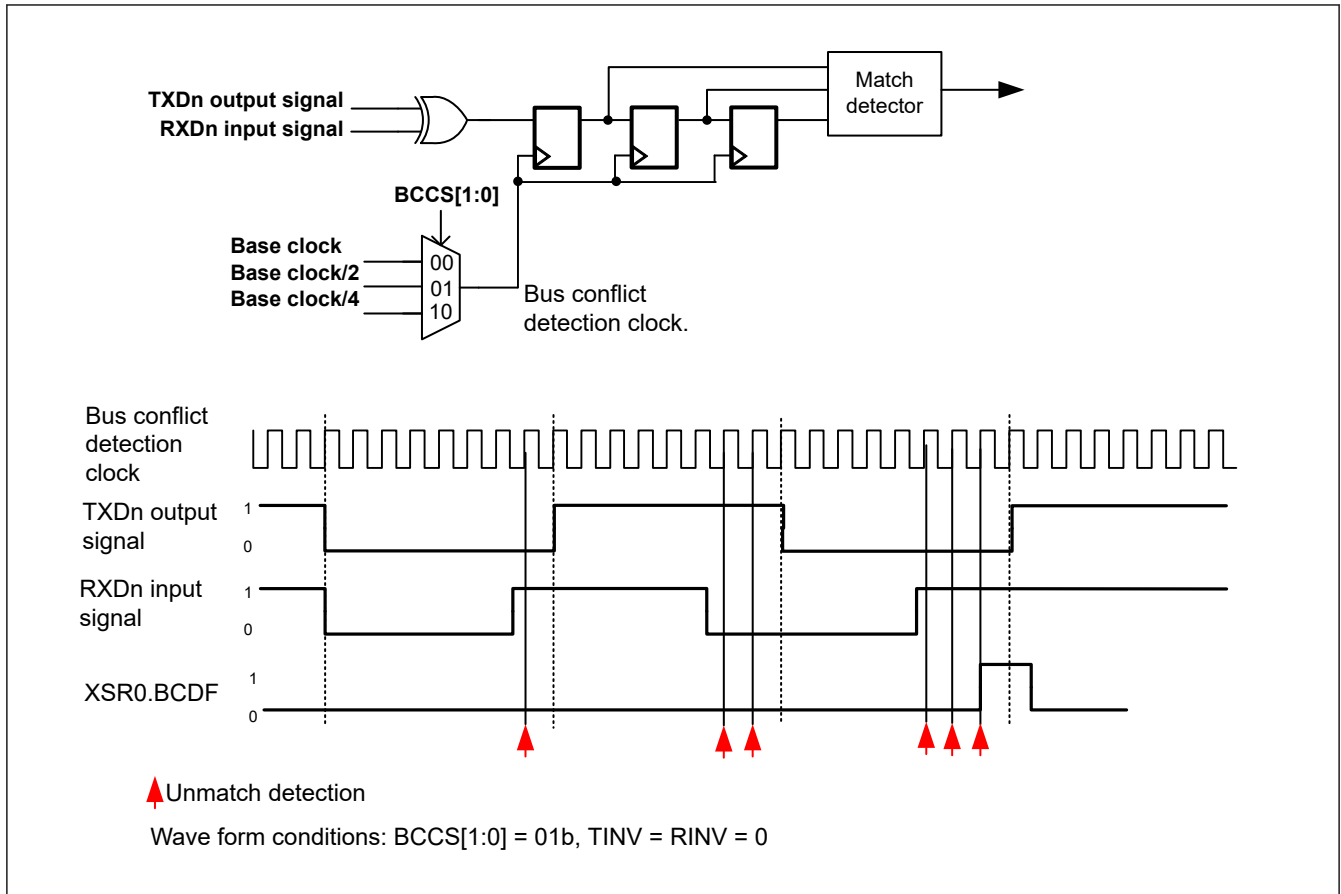


Figure 30.110 Operation Example of the Bus Conflict Detection Function

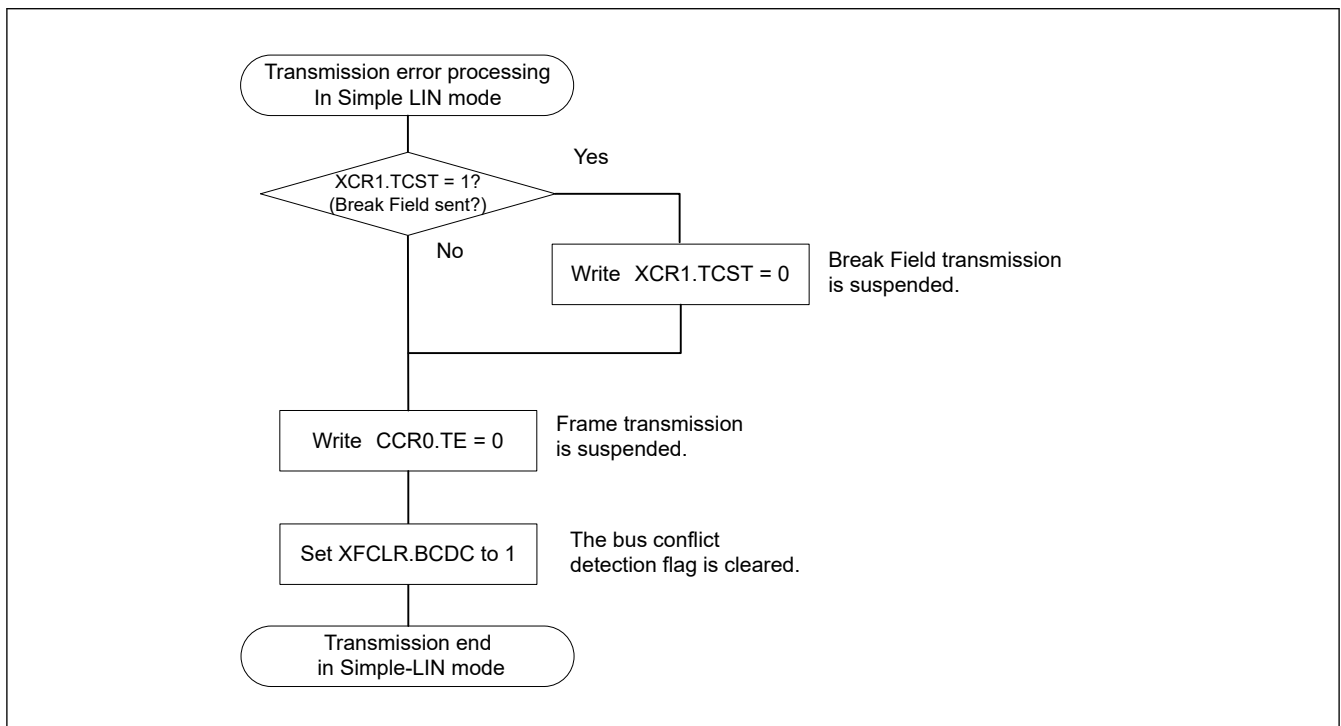


Figure 30.111 SCIn\_ERI interrupt handling flow at transmission in Simple LIN mode

### 30.11.4 Simple LIN Bit Rate Measurement Function

This function measures a bit rate between the effective edges of the input signal from the RXDn pin. Figure 30.112 shows an operation example of the bit rate measurement function.

1. Writing 1 to XCR1.SDST and XCR1.BMEN enables bit rate measurement. When this bit is set to 1, the valid edge interval of Control Field 0 and Control Field 1 data is measured. However, bit rate is not measured between the Break Field and the Break Delimiter. Set XCR1.BMEN and XCR1.SDST to 1 simultaneously, only when measuring bit rate.
2. Because bit rate is not measured in the Break Field, the effective edge detection flag is not set to 1 at the rising edge at the end of the Break Field, and the counter capture value is not stored in XSR1.TCNT[15:0].
3. The counter starts counting from the falling edge of the start bit in Control Field 0. The Break Delimiter count value is not captured in XSR1.TCNT[15:0].
4. The rising edge of the start bit is detected as an effective edge, and then the XSR0.AEDF flag is set to 1. If XCR0.AEDIE has been set to 1 at this time, an SCIn\_AED interrupt is output. The start bit count value is stored in XSR1.TCNT[15:0]. The XSR1.TCNT[15:0] value is retained until the effective capture value is read.
5. Even if an effective edge is input from the RXDn input pin, the count value of this effective edge timing is not captured because the XSR1.TCNT[15:0] value has not been read and retention has not been released. In this case, an SCIn\_AED interrupt is not output.
6. The XSR1.TCNT[15:0] value is read. Then the retention of XSR1.TCNT[15:0] is released and the XSR0.AEDF flag is cleared by hardware.
7. Because the retention of XSR1.TCNT[15:0] has been released, the count value is captured at the effective edge and is retained. At the same time, the XSR0.AEDF flag is set to 1, and if XCR0.AEDIE has been set to 1, an SCIn\_AED interrupt is output. The bit rate can be adjusted by calculating it from the count value between effective edges by software and by changing the SCI settings.
8. To disable bit rate measurement, write 0 to XCR1.BMEN.
9. The XSR0.AEDF value and the XSR1.TCNT[15:0] value remain unchanged at the effective edge timing because the bit rate measurement function is disabled.

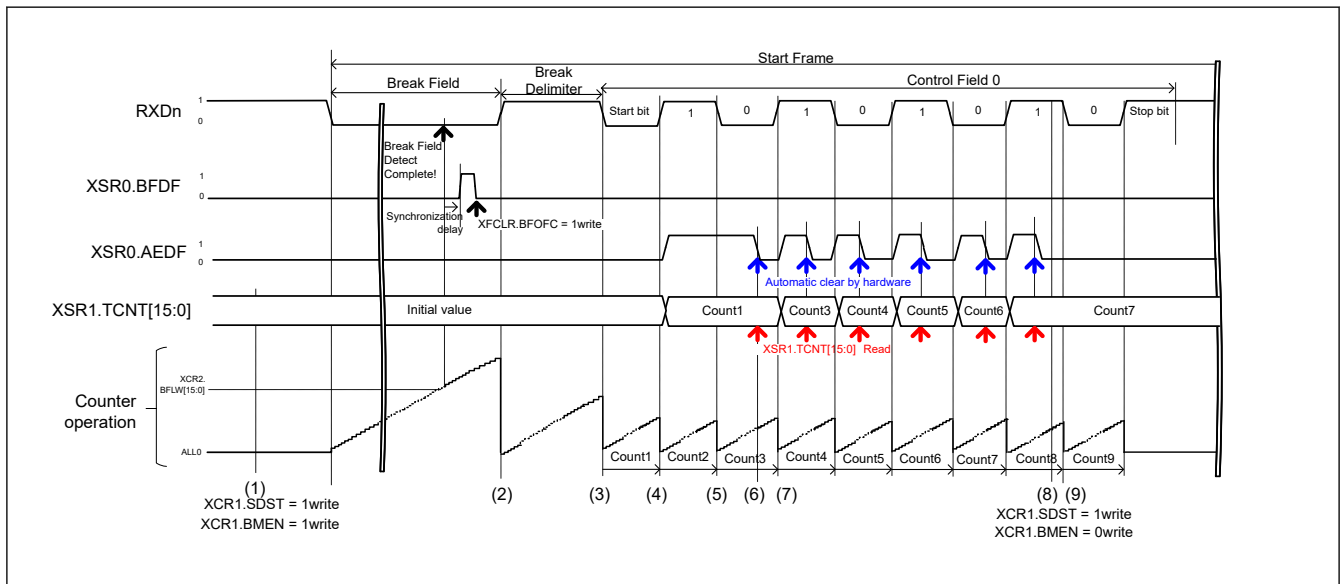


Figure 30.112 Operation Example of the Bit Rate Measurement Function

### 30.12 Interrupt Sources

#### 30.12.1 Buffer Operation for SCIn\_TXI and SCIn\_RXI Interrupts

If the conditions for an SCIn\_TXI and SCIn\_RXI interrupt are satisfied while the interrupt status flag in the ICU is 1, the ICU does not output the interrupt request but saves it internally with a capacity for retention of one request per source.

## 30.12.2 Interrupts in Asynchronous, Manchester, Clock Synchronous, and Simple SPI Modes

### (1) Non-FIFO selected

Table 30.43 lists interrupt sources in Asynchronous mode, Manchester mode, clock synchronous mode, and simple SPI mode.

A different interrupt vector can be assigned to each interrupt source. Individual interrupt sources can be enabled or disabled with the enable bits in the CCR0 register.

If the CCR0.TIE bit is 1, an SCIn\_TXI interrupt request is generated when transmit data is transferred from the TDR register to the TSR register. An SCIn\_TXI interrupt request can also be generated by using a single instruction to set the CCR0.TE and CCR0.TIE bits to 1 at the same time. An SCIn\_TXI interrupt request can activate the DTC or DMAC to handle data transfer.

An SCIn\_TXI interrupt request is not generated by setting the CCR0.TE bit to 1 when CCR0.TIE is 0 or by setting the CCR0.TIE bit to 1 when the CCR0.TE is 1.\*<sup>1</sup>

When new data is not written by the time of transmission of the last bit of the current transmit data and CCR0.TEIE is 1, the CSR.TEND flag is set to 1 and an SCIn\_TEI interrupt request is generated. Additionally, when CCR0.TE is 1, the CSR.TEND flag retains the value 1 until more transmit data is written to the TDR register, and setting CCR0.TEIE to 1 leads to the generation of an SCIn\_TEI interrupt request.

Writing data to the TDR register leads to clearing of the CSR.TEND flag and, after a certain time, discarding of the SCIn\_TEI interrupt request.

If the CCR0.RIE bit is 1, an SCIn\_RXI interrupt request is generated when received data is stored in the RDR register. An SCIn\_RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting any of the CSR.ORER, FER, PER or MSR.MER\*<sup>2</sup>, SYER\*<sup>2</sup>, PFER\*<sup>2</sup>, and SBER\*<sup>2</sup> flags to 1 when the CCR0.RIE bit is 1 leads to the generation of an SCIn\_ERI interrupt request.

An SCIn\_RXI interrupt request is not generated in this case. Clearing all these flags (ORER, FER, PER, MER\*<sup>2</sup>, SYER\*<sup>2</sup>, PFER\*<sup>2</sup> and SBER\*<sup>2</sup>) leads to discarding of the SCIn\_ERI interrupt request.

Note 1. To temporarily prohibit SCIn\_TXI interrupts on transmission of the last of the data when a new round of transmission is to be started, after handling the transmission-completed interrupt, control activation of the interrupt by using the interrupt request enable bit in the ICU rather than using the CCR0.TIE bit. This approach can prevent the suppression of SCIn\_TXI interrupt requests in the transfer of new data.

Note 2. MER, SYER, PFER, and SBER work as a factor of SCIn\_ERI interrupt only in Manchester mode. SYER, PFER, and SBER also only work if its enable bits (SYEREN, PFEREN, SBEREN in MECR) are set to "1".

### (2) FIFO selected

Table 30.44 lists interrupt sources in FIFO selected mode.

If the CCR0.TIE bit is 1, an SCIn\_TXI interrupt request is generated when the stored amount of data in the transmit-FIFO (TDR) register becomes the threshold value indicated in FCR.TTRG or below. An SCIn\_TXI interrupt request can also be generated by using a single instruction to set the CCR0.TIE and CCR0.TE bits to 1 simultaneously or by setting CCR0.TIE to 1 when CCR0.TE is 1.

An SCIn\_TXI interrupt request is not generated by setting CCR0.TE to 1 when CCR0.TIE is 0 or by setting the CCR0.TIE bit to 1 while the setting of the CCR0.TE bit is 1.

If CCR0.TEIE is 1 and if the next data is not written to the transmit-FIFO (TDR) register by the time the last bit of the transmit data is sent, the CSR.TEND flag is set to 1 and the SCIn\_TEI interrupt request is generated.

If CCR0.RIE is 1, the SCIn\_RXI interrupt request is generated when the stored amount of data in the transmit-FIFO (TDR) register is equal to or greater than the threshold value indicated in FCR.RTRG. When RTRG is 0, an SCIn\_RXI interrupt does not occur even when the amount of data in the receive FIFO is equal to 0.

If the CCR0.RIE bit is 1, when the CSR.ORER flag is set to 1 or data with a framing error or a parity error is stored in the transmit-FIFO (TDR) register, the SCIn\_ERI interrupt request is generated. When the amount of data stored in the transmit-FIFO (TDR) register is at the threshold value or above, the SCIn\_RXI interrupt request is also generated. The SCIn\_ERI interrupt request can be canceled, in which case CSR.ORER, FER, and PER flags are all cleared.

**Table 30.43** SCI interrupt sources with non-FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 4, 9)	Receive error	CSR.ORER, CSR.FER, CSR.PER, CSR.DFER, CSR.DPER, (MSR.MER, MSR.SYER, MSR.PFER, MSR.SBER)* <sup>1</sup>	CCR0.RIE	Not possible
SCIn_RXI (n = 0 to 4, 9)	Receive data full	CSR.RDRF	CCR0.RIE	Possible
	Address match	CSR.DCMF	CCR0.RIE	Possible
SCIn_TXI (n = 0 to 4, 9)	Transmit data empty	CSR.TDRE	CCR0.TIE	Possible
	TE = 0->1 detection			
SCIn_TEI (n = 0 to 4, 9)	Transmit end	CSR.TEND	CCR0.TEIE	Not possible

Note: Only SCI0 supports Manchester mode.

Note 1. MER, SYER, PFER, and SBER work as a factor of SCIn\_ERI interrupt only in Manchester mode. SYER, PFER, and SBER also only work if its enable bits (SYEREN, PFEREN, SBEREN in MCR) are set to 1.

**Table 30.44** SCI interrupt sources with FIFO selected

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 4, 9)	Receive error	CSR.ORER, CSR.FER, CSR.PER, CSR.DFER, CSR.DPER	CCR0.RIE	Not possible
		FRSR.DR (when FCR.DRES = 1)	CCR0.RIE	Not possible
SCIn_RXI (n = 0 to 4, 9)	Receive data full	CSR.RDRF	CCR0.RIE	Possible
	Receive data ready	FRSR.DR (when FCR.DRES = 0)	CCR0.RIE	Possible
	Address match	CSR.DCMF	CCR0.RIE	Possible
SCIn_TXI (n = 0 to 4, 9)	Transmit data empty	CSR.TDRE	CCR0.TIE	Possible
	TE=0->1 detection			
SCIn_TEI (n = 0 to 4, 9)	Transmit end	CSR.TEND	CCR0.TEIE	Not possible

Note: Only SCI0 supports Manchester mode.

### 30.12.3 Interrupts in Smart Card Interface Mode

Table 30.45 lists interrupt sources in smart card interface mode. A transmit end interrupt (SCIn\_TEI) request and an address match (SCIn\_AM) request cannot be used in this mode.

**Table 30.45** SCI Interrupt sources in Smart Card Interface Mode

Name	Interrupt source	Interrupt flag	Interrupt enable	DTC or DMAC activation
SCIn_ERI (n = 0 to 4, 9)	Receive error or error signal detection	CSR.ORER, CSR.PER, CSR.ERS	CCR0.RIE	Not possible
SCIn_RXI (n = 0 to 4, 9)	Receive data full	CSR.RDRF	CCR0.RIE	Possible
SCIn_TXI (n = 0 to 4, 9)	Transmit data empty	CSR.TEND	CCR0.TIE	Possible
	When set TE = 0->1			

Data transmission or reception using the DTC or DMAC is also possible in smart card interface mode, similar to normal SCI mode. In transmission, when the CCR0.TEND flag is set to 1, an SCIn\_TXI interrupt request is generated. This SCIn\_TXI interrupt request activates the DTC or DMAC, allowing transfer of transmit data if the SCIn\_TXI request is previously specified as a source of DTC or DMAC activation. The TEND flag is automatically set to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept at 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified



number of bytes, including retransmission after an error occurrence. However, the CSR.ERS flag is not automatically set to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the CCR0.RIE bit to 1 to enable an SCIn\_ERI interrupt request to be generated at error occurrence.

When transmitting or receiving data using the DTC or DMAC, always enable the DTC or DMAC before making the SCI settings. For DTC or DMAC settings, see [section 17, Data Transfer Controller \(DTC\)](#), [section 16, DMA Controller \(DMAC\)](#).

In reception, an SCIn\_RXI interrupt request is generated when receive data is set to the RDR register. This SCIn\_RXI interrupt request activates the DTC or DMAC, allowing transfer of the receive data if the SCIn\_RXI request is previously specified as a source of DTC or DMAC activation. If an error occurs, the error flag is set. Therefore, the DTC or DMAC is not activated and an SCIn\_ERI interrupt request is issued to the CPU instead. The error flag must be cleared.

### 30.12.4 Interrupts in Simple IIC Mode

[Table 30.46](#) lists the interrupt sources in simple IIC mode. The STI interrupt is allocated to the transmit end interrupt (SCIn\_TEI) request. The receive error interrupt (SCIn\_ERI) and the address match (SCIn\_AM) request cannot be used.

The DTC or DMAC can also be used to handle transfer in simple IIC mode.

When the ICR.IICINTM bit is 1:

- An SCIn\_RXI request is generated on the falling edge of the SCLn signal for the 8<sup>th</sup> bit. If SCIn\_RXI is previously set up as an activation source for the DTC or DMAC, the SCIn\_RXI request activates the DTC or DMAC to handle transfer of the received data.
- An SCIn\_TXI request is generated on the falling edge of the SCLn signal for the 9<sup>th</sup> bit (acknowledge bit). If SCIn\_TXI is previously set up as an activation source for the DTC or DMAC, the SCIn\_TXI request activates the DTC or DMAC to handle transfer of the transmit data.

When the ICR.IICINTM bit is 0:

- An SCIn\_RXI request (ACK detection) is generated if the input on the SDAn pin is low on the rising edge of the SCLn signal for the 9<sup>th</sup> bit (acknowledge bit)
- An SCIn\_TXI request (NACK detection) is generated if the input on the SDAn pin is high on the rising edge of the SCLn signal for the 9<sup>th</sup> bit (acknowledge bit)
- If SCIn\_RXI is previously set up as an activation source for the DTC or DMAC, the SCIn\_RXI request activates the DTC or DMAC to handle transfer of the received data.

If the DTC or DMAC is used for data transfer in reception or transmission, always set up and enable the DTC or DMAC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in ICR are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

**Table 30.46 SCI interrupt sources in Simple IIC Mode**

Name	Interrupt source		Interrupt flag	Interrupt enable	DTC or DMAC activation
	ICR.IICINTM = 1	ICR.IICINTM = 0			
SCIn_RXI (n = 0 to 4, 9)	Reception end	—	—	CCR0.RIE	Possible* <sup>1</sup>
	—	ACK detection	—		Possible
SCIn_TXI (n = 0 to 4, 9)	Transmission end	—	—	CCR0.TIE	Possible* <sup>1</sup>
	—	NACK detection	—		Possible
SCIn_TEI(STIn) (n = 0 to 4, 9)	Completion of generation of a start, restart, or stop condition		ICR.IICSTIF	CCR0.TEIE	Not possible

Note 1. If the DMAC or DTC are being used, you cannot confirm whether ACK or NACK.

### 30.12.5 Interrupts in Simple LIN mode

[Table 30.47](#) lists interrupt sources in Simple LIN mode.

**Table 30.47 SCI interrupt sources in Simple LIN mode**

Name	Interrupt Sources	Interrupt Flag	Flag the needs to be confirmed	Interrupt Enable	DTC/DMAC Activation
SCIn_ERI (n = 0, 1)	Receive error	CSR.ORER, CSR.FER, CSR.PER	—	CCR0.RIE	Not Possible
		XSR0.BCDF		XCR0.BCDIE	
		XSR0.COF		CCR0.RIE, XCR0.COFIE	
SCIn_RXI(n = 0, 1)	Receive data full flag	CSR.RDRF	XSR0.CF0MF XSR0.CF1MF XSR0.PIBDF	CCR0.RIE	XSR0.SFSF = 0: Possible XSR0.SFSF = 1: Not Possible
SCIn_AED (n = 0, 1)	Active edge detection	XSR0.AEDF	—	XCR0.AEDIE	Possible
SCIn_TXI(n = 0, 1)	Transmit data empty interrupt	CSR.TDRE	—	CCR0.TIE	Possible
	When set TE = 0->1				
	Break Field output completion	XSR0.BFOF		CCR0.TIE, XCR0.BFOIE	
SCIn_TEI(n = 0, 1)	Transmit end	CSR.TEND	—	CCR0.TEIE	Not Possible
SCIn_BFD(n = 0, 1)	Break Field Detection	XSR0.BFDF	—	XCR0.BFDIE	Not Possible (Unnecessary)

In Simple LIN mode, in addition to reception errors (ORER, FER, PER), an SCIn\_ERI interrupt request is output when a bus conflict is detected during transmission, or when a counter overflow of the Simple LIN module occurs. At this time, a SCIn\_RXI interrupt request is not output. The SCIn\_ERI interrupt request can be canceled by clearing all the flags.

When transmitting Start Frame, if CCR0.TIE = 1 and XCR0.BFOIE = 1, a SCIn\_TXI interrupt request is output when Break Field transmission is completed. When Control Field 0 data is written to the TDR register, data transmission starts. Therefore, transmission using DTC or DMAC is possible.

Set CCR0.TEIE = 1 after writing the last transmit data to the TDR register and transmission starts.

During Start Frame reception (XSR0.SFSF = 1), reception using DTC or DMAC by SCIn\_RXI interrupt is not possible. Check the CSR register and XSR0 register, check the reception status (See [Figure 30.83](#)), and then clear the flag. Also read the RDR register (if you do not need to check the received data value, clear the RDRF flag without reading the RDR register). When reception of Control Field 1 is completed (XSR0.CF1MF = 1), Start Frame detection is disabled (XSR0.SFSF = 0) and reception using DTC or DMAC is possible. Be sure to read the RDR register.

When Start Frame / Break Field detection is enabled (XCR1.SDST = 1), if a Break Field longer than the period set in XCR2.BFLW [15:0] is received, the BFDF flag is set and a SCIn\_BFD interrupt request is output. Then SCI becomes the Start Frame reception state. Clear the BFDF flag.

When Start Frame / Break Field detection is enabled (XCR1.SDST = 1) and the bit rate measurement function is enabled (XCR1.BMEN = 1), an SCIn\_AED interrupt factor is output when an active edge is detected. Read the timer count capture value (XSR1.TCNT [15:0]).

### 30.13 Event Linking

By using interrupt request signals as event signals, the SCIn can provide linked operation through the ELC for modules selected in advance.

Event signals can be output regardless of the values of the associated interrupt request enable bits.

#### (1) Error event output (receive error or error signal detected) (SCIn\_ERI, n = 0 to 4, 9)

- Indicates abnormal termination because of a parity error during reception in Asynchronous mode
- Indicates abnormal termination because of a framing error during reception in Asynchronous mode
- Indicates abnormal termination because of an overrun error during reception
- Indicates abnormal termination due to a Manchester error during reception (Only in Manchester mode).

- Indicates that a preface error occurred upon reception and abnormal termination occurred (only in Manchester mode and MCR.PFEREN = 1).
- Indicates that a start bit error occurred during reception and abnormal termination occurred (only in Manchester mode and MCR.SBEREN = 1).
- Indicates that a reception sync error occurred during reception and abnormal termination occurred (only in Manchester mode and only when MCR.SYEREN = 1).
- Indicates detection of the error signal during transmission in smart card interface mode
- The CSR.FER and PER flags are 0, and receive data less than the receive FIFO data trigger number is set in a reception FIFO buffer, and it indicates that 15 etus elapse when FIFO is selected and the FCR.DRES bit is 1
- In Simple LIN mode, indicates that the 16-bit counter in the Simple LIN module has overflowed.
- In Simple LIN mode, a bus collision is detected during transmission (CCR0.TE = 1).

#### (2) Receive data full event output (SCIn\_RXI, n = 0 to 4, 9)

- Indicates that ACK is detected if the ICR.IICINTM bit is 0 in simple IIC mode
- Indicates that the 8th-bit SCLn falling edge is detected if the ICR.IICINTM bit is 1 in simple IIC mode
- When the ICR.IICINTM bit is 1 during master transmission in simple IIC mode, set the ELC so that receive data full events are not used

#### Non-FIFO selected

- Indicates that received data is set in the Receive Data Register (RDR).

#### FIFO selected

- Using this event output is prohibited.

#### (3) Transmit data empty event output (SCIn\_TXI, n = 0 to 4, 9)

- Indicates that the CCR0.TE bit is changed from 0 to 1
- Indicates that transmission is complete in smart card interface mode
- Indicates that NACK is detected if the ICR.IICINTM bit is 0 in simple IIC mode
- Indicates that the 9th-bit SCLn falling edge is detected if the ICR.IICINTM bit is 1 in simple IIC mode
- In Simple LIN mode, indicates that Break Field output is complete.

#### Non-FIFO selected

- Indicates that transmit data is transferred from the Transmit Data Register (TDR) to the Transmit Shift Register (TSR).

#### FIFO selected

- Using this event output is prohibited.

#### (4) Transmit end event output (SCIn\_TEI, n = 0 to 4, 9)

- Indicates the completion of transmission
- Indicates that the starting condition, resumption condition, or termination condition is generated in simple IIC mode
- In Smart Card mode, the transmit end event is not output.

Note: When FIFO is selected, using this event output is prohibited

#### (5) Address match event output (SCIn\_AM, n = 0 to 4, 9)

- Indicates a match of the comparison data (CCR4.CMPD) with one frame of receive data when CCR0.DCME is set to 1 in Asynchronous mode, including multi-processor mode.

## (6) Active edge detection event output (SCIn\_AED, n= 0, 1)

- In Simple LIN mode, when CCR1.BMEN is 1, it indicates that a valid edge has been detected in the RXD input signal.

## 30.14 Noise Cancellation Function

Figure 30.113 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a 2-stage flip-flop circuit and a match detection circuit. When the input signals of the noise filter and the output signals of the 2-stage flip-flop circuits completely match, the matched level is conveyed as an internal signal. Unless otherwise matched, the previous value is retained. When the same level is retained for 3 cycles or longer on the sampling clock of the noise filter, it is considered as a valid receive signal. A change in pulse for 3 cycles or shorter is considered as noise, not as a receive signal. (When CCR1.NFM = 0)

When CCR2.ABCSE2 = 1 in asynchronous mode, the above noise cancellation cannot be used because one bit has only 4 base clocks. Therefore, only in this case, set the noise cancellation mode to transmit the majority value internally by a majority vote of 3 sampling values instead of matching 3 consecutive sampling values. (CCR1.NFM = 1)

In Asynchronous mode, Manchester and Simple LIN modes, the noise cancellation function can be applied to the receive signal input to the RXDn pin. The sampling period of the noise filter can be selected from the base clock period and the divided clock of the baud rate generator clock source by CCR1.NFCS[2:0].

- When CCR1.NFCS[2:0] = 000b, CCR2.ABCS = 0 and CCR2.ABCSE = 0, the cycle is 1/16 of a 1-bit period.
- When CCR1.NFCS[2:0] = 000b, CCR2.ABCS = 1 and CCR2.ABCSE = 0, the cycle is 1/8 of a 1-bit period.
- When CCR1.NFCS[2:0] = 000b, CCR2.ABCSE = 1, the cycle is 1/6 of a 1-bit period.
- When CCR1.NFCS[2:0] = 000b, CCR2.ABCSE2 = 1, the cycle is 1/4 of a period 1 transfer bit.
- For the CCR1.NFCS value, select “000b” or “001b” if ABCSE = 1 or ABCSE2 = 1. At least, the sampling period of Noise Filter should not be greater than the base clock due to the CCR2.BRR setting value.

In simple IIC mode, this function can be used for each input on SDAn and SCLn. The sampling clock is selected from divided clock of baud rate generator settings by CCR1.NFCS[2:0].

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When CCR0.TE and CCR0.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, the function determines that a level match is detected and the result is conveyed as an internal signal. If the input data is 0, the Noise Filter output holds its initial value until all three sampled pin levels match (CCR1.NFM = 0) or two or more match (CCR1.NFM = 1).

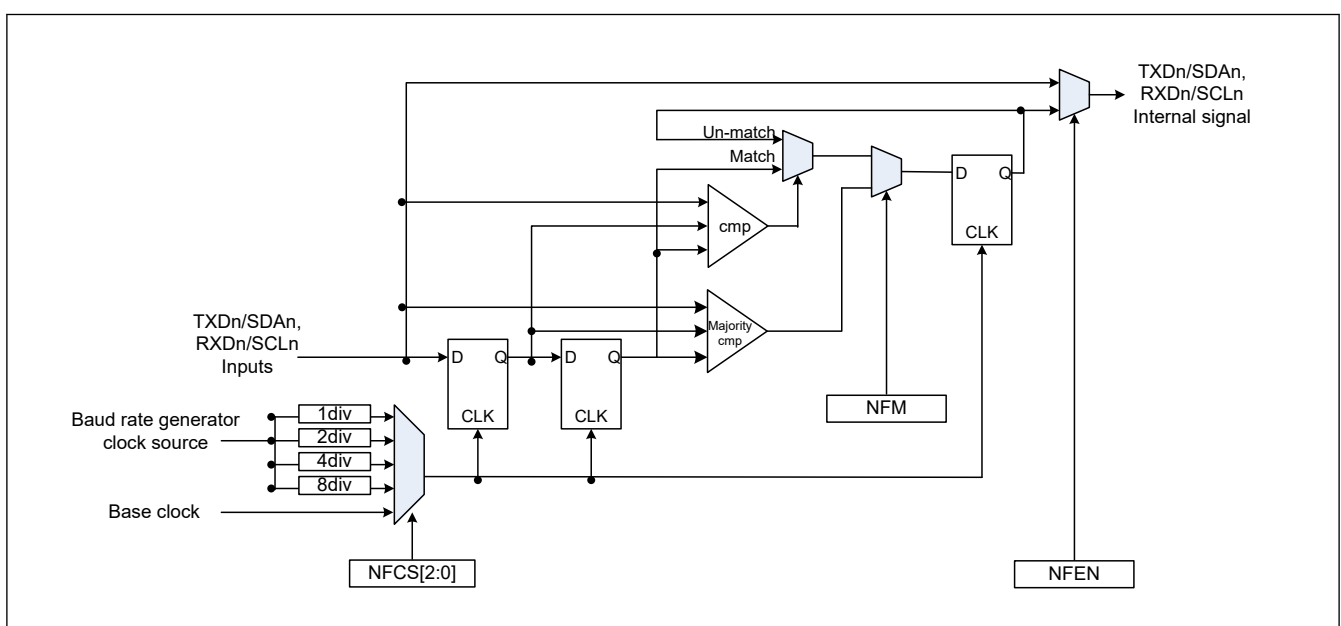


Figure 30.113 Digital noise filter circuit block diagram

### 30.15 RS-485 Driver Control Function

Setting the DEN bit in the SCI common control register3 (CCR3) to 1 enables the RS-485 driver control function and generates a DEN (Driver Enable) signal that enables the external transceiver transmission mode. The DEN signal outputs a valid level for the period with driver assertion time and driver negate time added before and after data transmission. The DEN signal valid level is set by the DEPOL bit in the driver control register (DCR).

The driver assertion time is the time from when the DEN signal is valid until the start bit starts. Set by DEAST [4:0] of driver control register (DCR).

The driver negate time is the time from the end of the last stop bit of the transmitted message to the invalidation of the DEN signal. Set with DENG [4:0] of the driver control register (DCR).

DEAST and DENG are expressed in base clock period (1/4, 1/6, 1/8, or 1/16 bit period, see Table 30.6). For details, see section 30.2.13. DCR : Driver Control Register.

When this function is used (CCR3.DEN = 1), the CSR.TEND set timing and SCIn\_TEI interrupt output timing are at the end of the driver negation time.

When transmission is completed and the next transmission data is not written before the DEN signal is negated, the DEN signal is negated once. If the timing for writing the next transmit data is not in time, assert the DEN signal after negating it again, insert the driver assertion time, and transmit the next data. If you want to perform the next transmission with the DEN signal asserted, write the next transmission data to the TDR quickly enough in consideration of the synchronization delay time of the register.

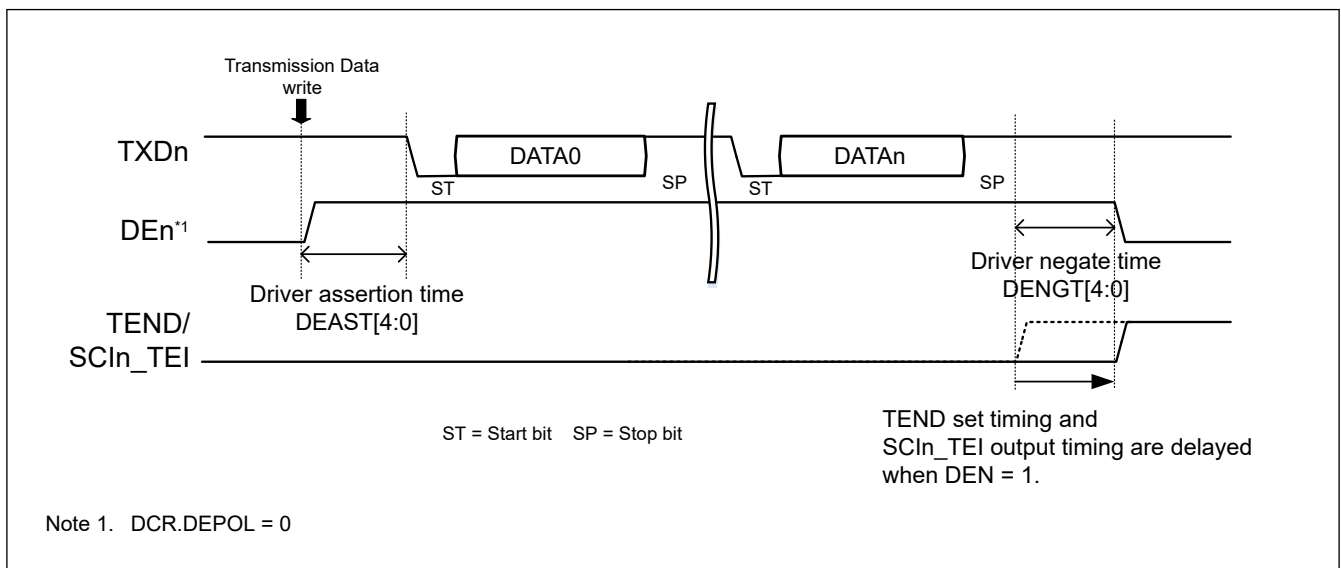


Figure 30.114 The image waveform for RS-485 driver control DE signal output

### 30.16 Loopback Function

The loopback function can be used in Asynchronous mode with the internal clock, and Manchester mode with the internal clock, and Clock synchronous mode with the internal clock.

When 1 is written to the SPLP bit in the CCR1 register, SCI blocks the external input (RXDn) path and connects the output path of the transmit data register and the input path of the receive data register.

When this function is used with TINV bit = 1, inversion of transmission data becomes reception data. However, this function can be used with TINV = 1 only when operating in clock synchronous mode internal clock.

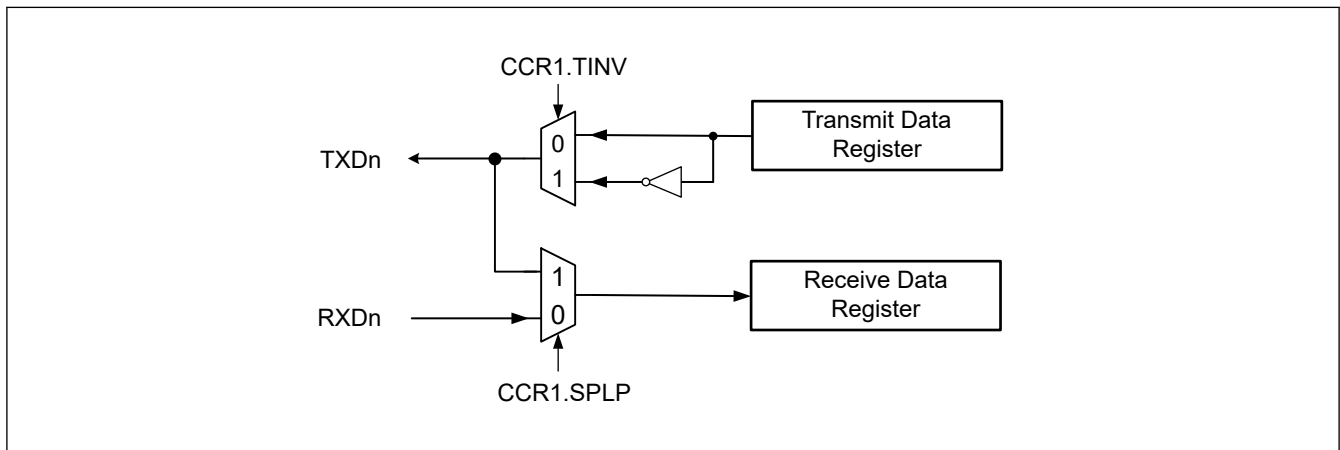
Table 30.48 shows the relationship between the TINV and SPLP bit settings and the received data.

**Table 30.48 TINV and SPLP bit settings and received data**

CCR1.TINV	CCR1.SPLP	Receive Data	Communication mode		
			Asynchronous	Manchester	Clock synchronous
			internal clock	internal clock	internal clock
—	0	Receive Data from RXDn pin	Possible	Possible	Possible
0	1	Transmit Data	Possible	Possible	Possible
1	1	Inverted transmit data	Impossible	Impossible	Possible

Note: —: don't care

Figure 30.115 shows the configuration of the shift register input / output path in loopback mode.



**Figure 30.115 Shift register input output configuration image in loopback mode**

### 30.17 Half-Duplex communication Function

Do not use the half-duplex communication function in Simple IIC, Simple SPI and Smart Card Interface modes.

In other communication modes, if the CCR1.SHARPS bit is set to 1, half-duplex communication using the TXDn pin is possible. When half-duplex communication is used, transmission and reception must be performed exclusively. Transmission and reception settings (CCR0.TE = 1 and CCR0.RE = 1) is prohibited.

However, if half-duplex communication is performed as the master reception in clock synchronous mode, perform transmission / reception settings (CCR0.TE = 1 and CCR0.RE = 1) and perform dummy transmission. By dummy transmission (arbitrary transmission data is written to TDR), SCKn is output and reception is enabled. The dummy transmission data is discarded inside the SCI and is not actually transmitted.

During half-duplex communication, only the TXDn pin is used as the communication pin. Output when CCR0.TE = 1, input when CCR0.TE = 0.

### 30.18 Synchronizer Bypass Function

The SCI has a bus clock (PCLK) and the operation clock (TCLK). And these have each operating circuit. Therefore, there is a synchronization circuit for signal transfer between different clocks, and synchronization delay time is required for signal propagation between different clocks.

However, the synchronization circuit can be bypassed by the CCR3.BPEN bit only when the same clock is input to the bus clock and the operation clock. In this case, eliminates synchronization delay time and improves responsiveness. Figure 30.116 shows the image waveform of the bypass function.

The SCI also has a synchronization circuit between the communication clock (SCKn) and the operation clock (TCLK), but this synchronization circuit cannot be bypassed.

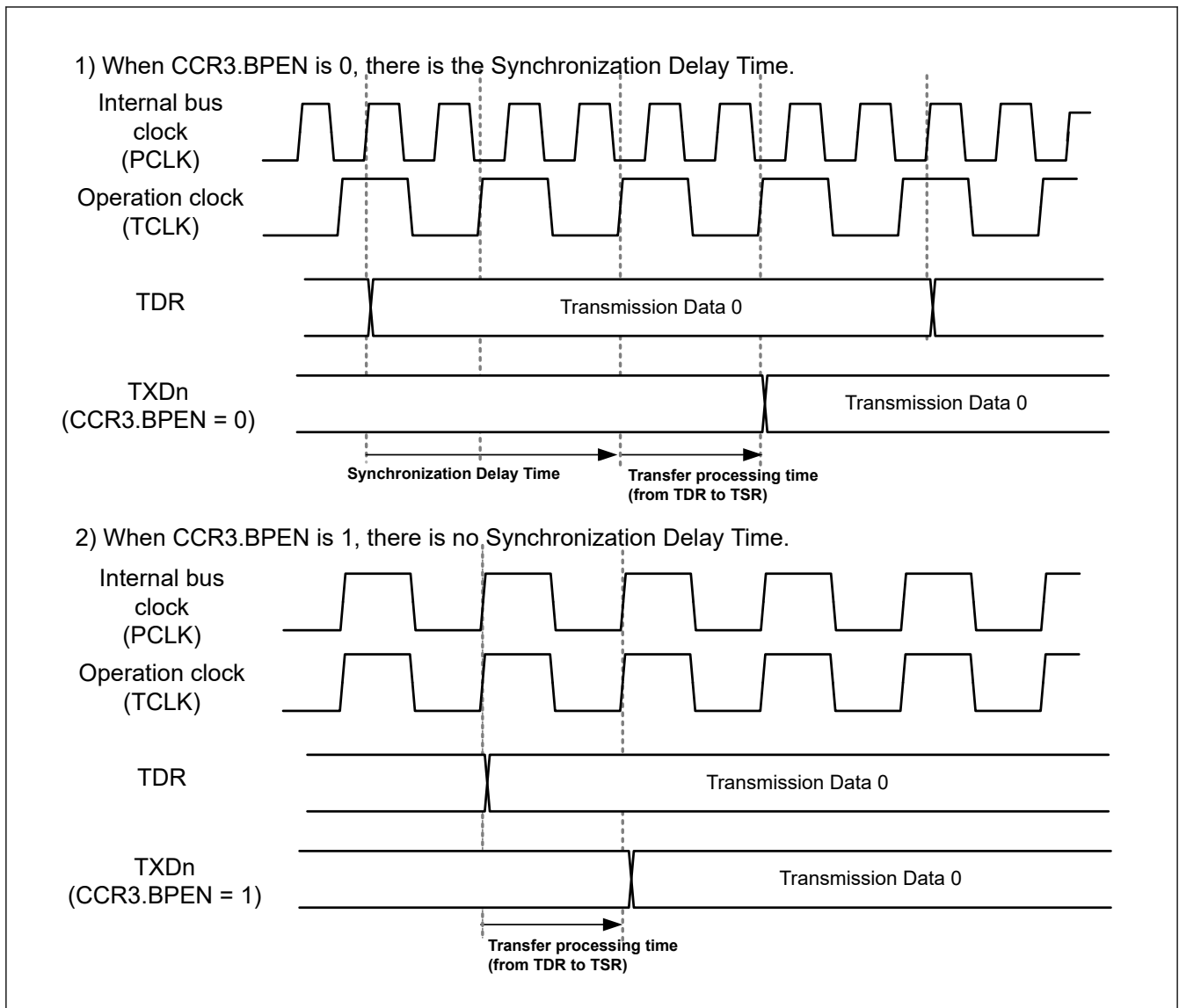


Figure 30.116 Image waveform of Synchronizer bypass function

## 30.19 Usage Notes

### 30.19.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable SCI operation. The SCI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

### 30.19.2 SCI Operation during Low Power State

#### (1) Transmission

Before using the power consumption reduction function to reduce SCI's power consumption, do the following to confirm transmission end (CSR.TEND = 1):

- Set the output pin state after transmission operation is stopped by CCR1.SPB2DT, SPB2IO.
- Stop the transmission (CCR0.TIE = 0, TE = 0, TEIE = 0)

When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same operation mode after cancellation of the low power consumption state, set the TE bit to 1, read CSR, and write data to TDR sequentially to start data transmission. To transmit data with a different operation mode, initialize the SCI first.

To start transmission using the DMAC or DTC after cancellation from software standby mode, set the CCR0.TE and CCR0.TIE bit to at the same time. Then SCIn\_TXI interrupt flag is generated, which causes the DMAC or DTC to write the transmit data, which starts transmission.

Figure 30.117 shows a sample flowchart for transition to software standby mode during transmission. Figure 30.118 and Figure 30.119 show the port pin states during transition to software standby mode.

(2) Reception

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (CCR0.RE = 0). If transition is made during data reception, the data being received will be invalid.

Figure 30.120 shows a sample flowchart for reception to software standby mode during reception.

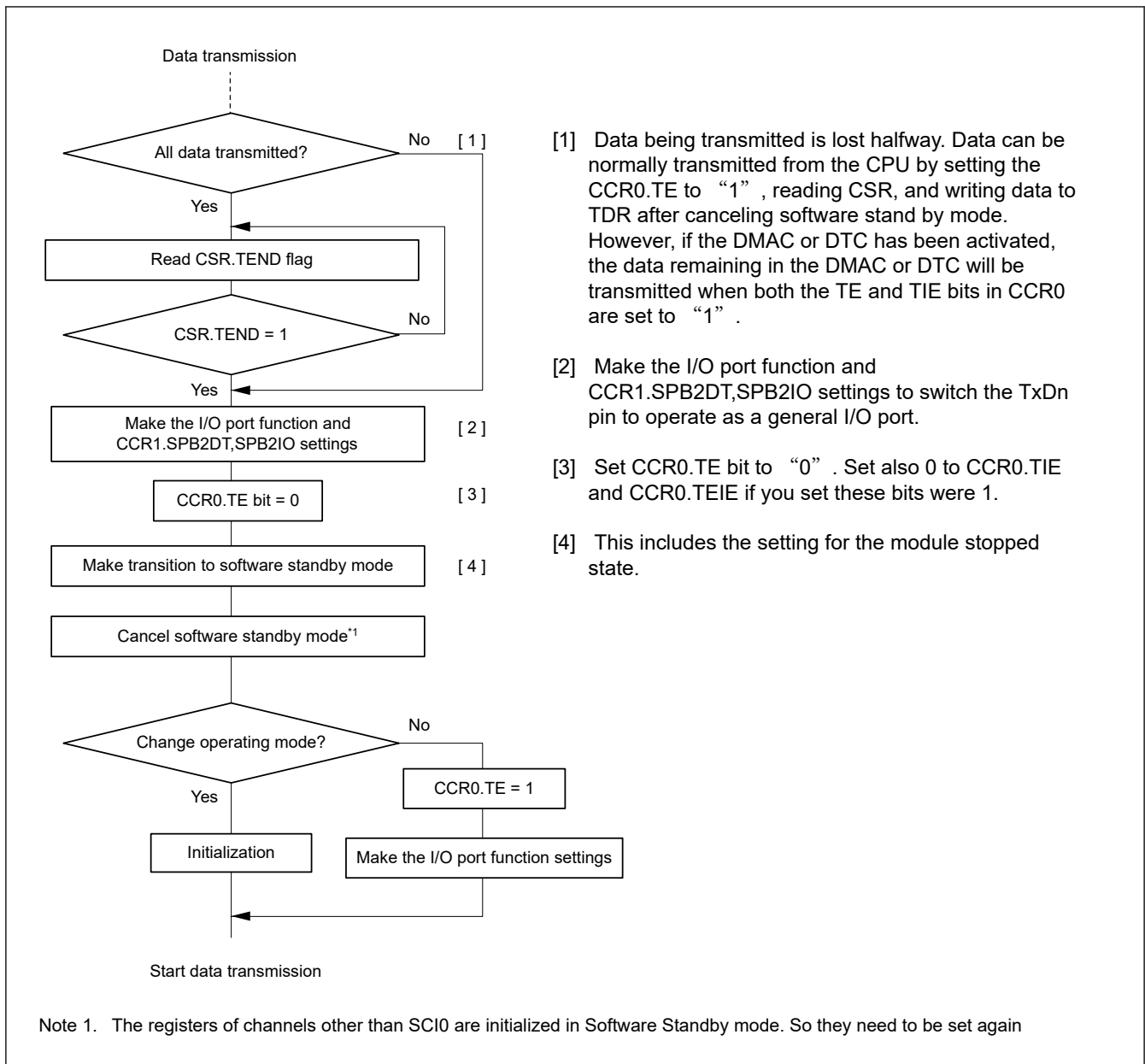
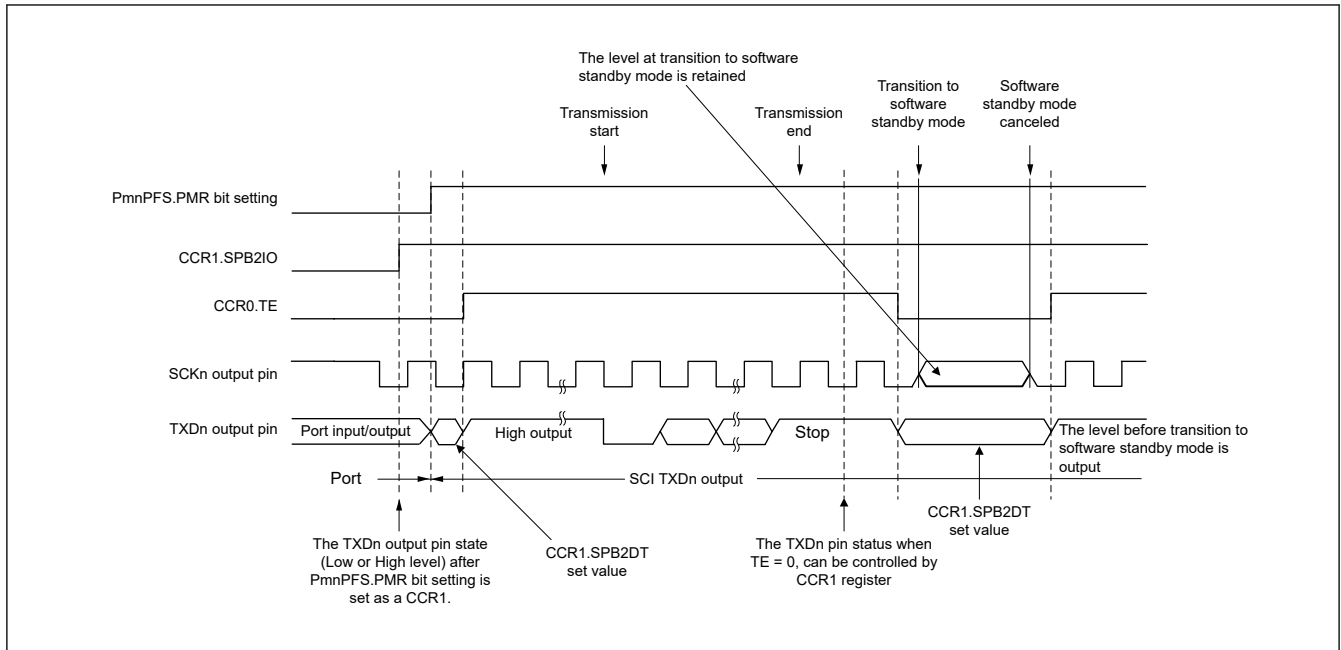
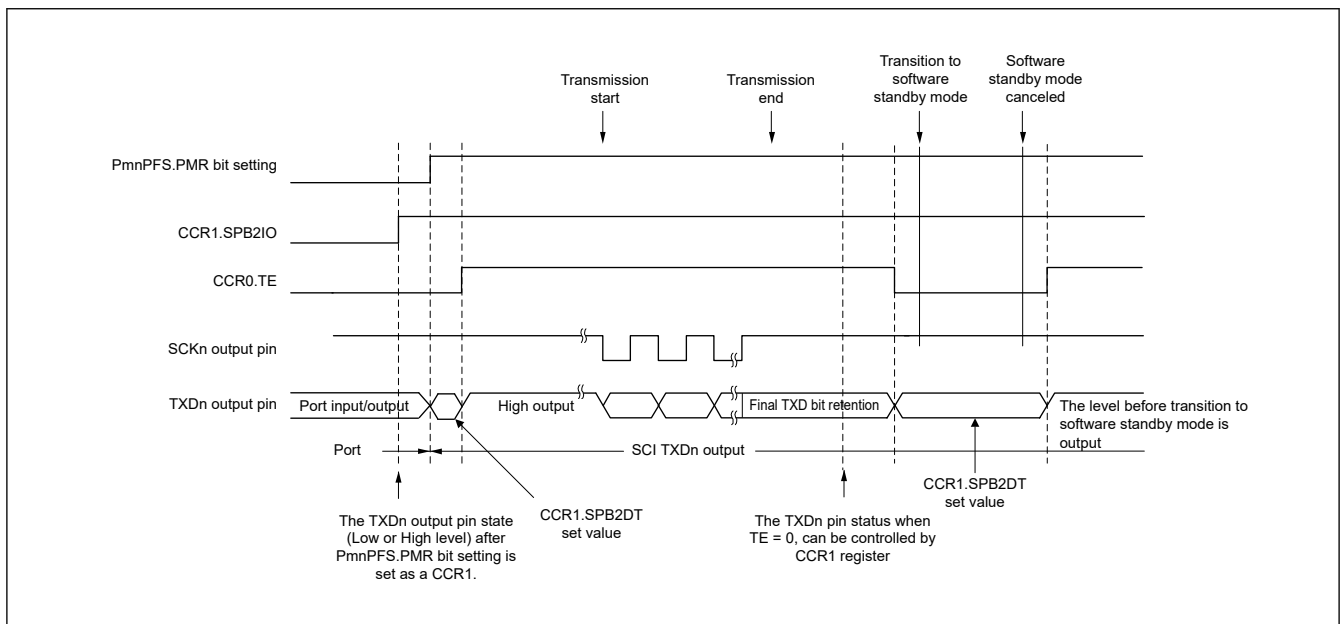


Figure 30.117 Example of Flowchart for Transition to Software Standby Mode during Transmission





**Figure 30.118 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)**



**Figure 30.119 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)**

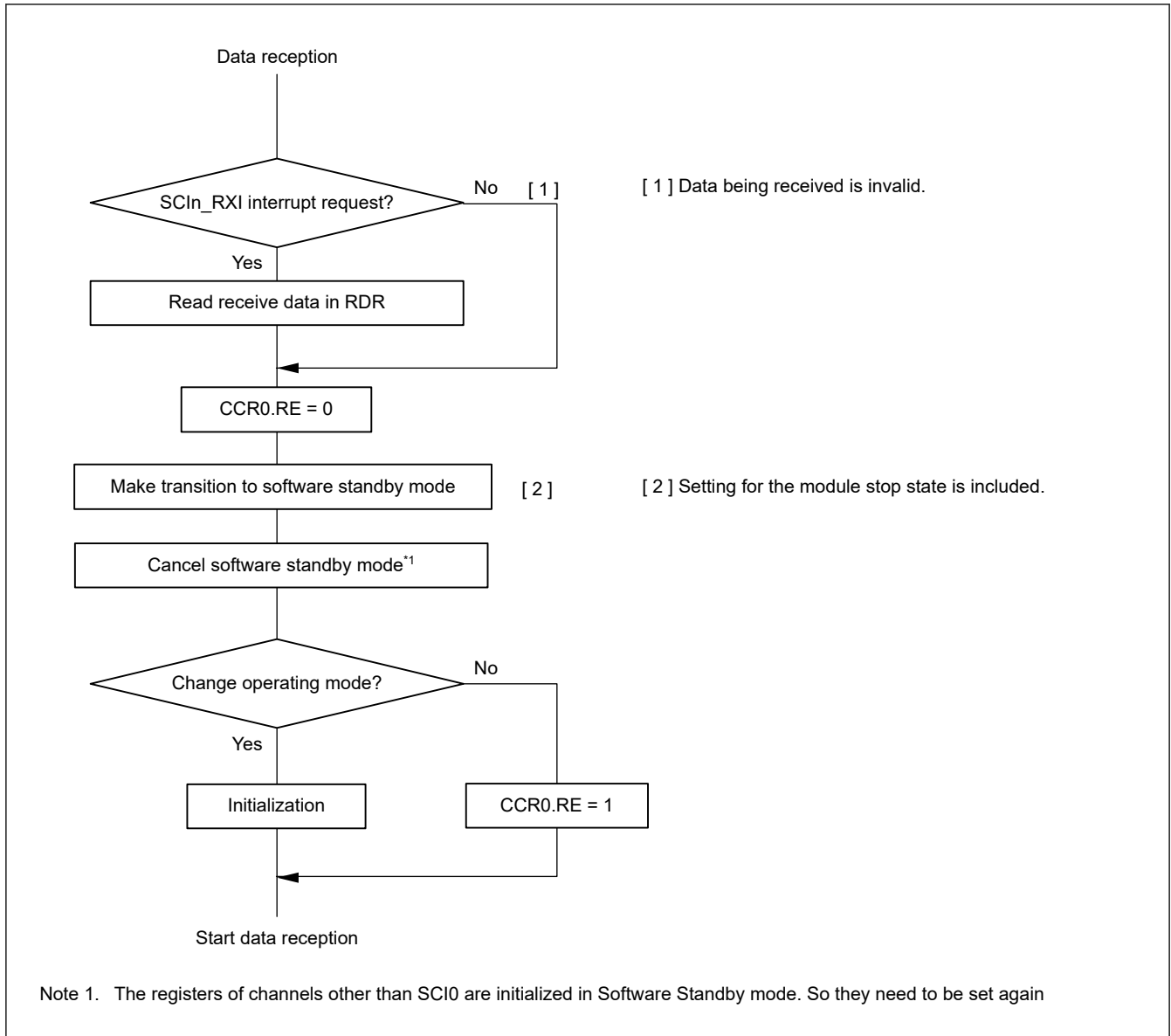


Figure 30.120 Example of Flowchart for Reception to Software Standby Mode during Reception

### 30.19.3 Break Detection and Processing

#### (1) Non-FIFO selected

When a framing error is detected, a break can be detected by reading CSR.RXDMON bit value. In a break, the input from the RXDn pin becomes all 0s, and the CSR.FER flag is set to 1 to indicate a framing error, and the CSR.PER flag might also be set to 1 to indicate a parity error. The SCI continues the receive operation even after a break is received. Therefore, even if the FER flag is 0, indicating that no framing error occurred, it is set to 1 again. When the CCR3.RXDESEL bit is 1, the SCI sets the CSR.FER flag to 1 and stops receiving operations until a start bit of the next data frame is detected. If the CSR.FER flag is set to 0, the CSR.FER flag retains 0 during the break.

When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit on the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

#### (2) FIFO selected

After a framing error is detected and when the SCI detects that continuous receive data is 0 for 1 frame, reception stops. When a framing error is detected, a break can be detected by reading the CSR.RXDMON flag value. After the RXDn signal is in high and the break is finished, data reception to the receive-FIFO (RDR) register resumes.

### 30.19.4 Mark State and Production of Breaks

When the CCR0.TE bit is 0, disabling serial transmission, the state of the TXDn pin can be set using the CCR1.SPB2IO and CCR1.SPB2DT bits. With this approach, a TXDn pin can be placed in the mark state to transmit a break.

Before setting the CCR0.TE bit to 1, enabling serial transmission, set the SPB2IO and SPB2DT bits to put the communication line in the mark state (the state of 1), and change the TXDn pin using I/O port function. To output a break on data transmission, after setting the TXDn pin to output 0 by setting the SPB2IO and SPB2DT bits, change the TXDn pin using the I/O port function and set the CCR0.TE bit to 0. When the CCR0.TE bit is set to 0, the transmitter is initialized regardless of the current state of transmission.

### 30.19.5 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission can be start by writing transmit-data to TDR even if CSR.ORER is 1. However, reception cannot be started. Note also that the receive error flags cannot be set to 0 even if the CCR0.RE is set to 0 (serial reception is disabled).

### 30.19.6 Writing Data to TDR

#### (1) Non-FIFO selected

Data can be written to TDR anytime when CCR0.TE is 1. However, if new data is written to TDR when transmit data is remaining in TDR, the previous data in TDR is lost because it has not been transferred to TSR yet. If you use DTC or DMAC, be sure to write transmit data to TDR in the SCIn\_TXI interrupt request handling routine.

#### (2) FIFO selected

Data can be written to transmit-FIFO(TDR) when CCR0.TE is 1. Check the number of writable data with the FTSR.T [5:0] bit.

### 30.19.7 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

#### (1) Start of transmission

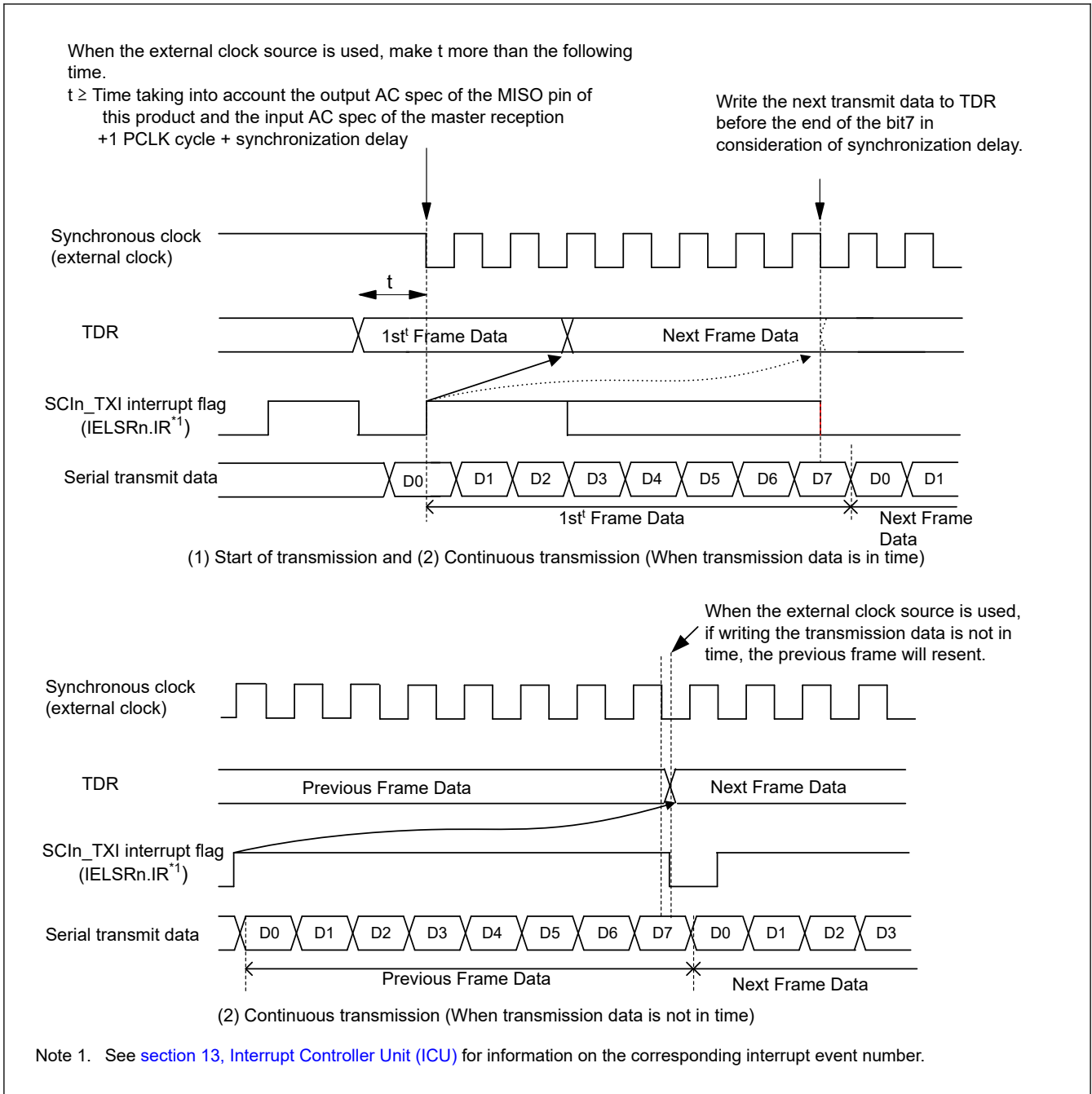
Update TDR by the CPU, DMAC, or DTC and wait at least the following time until the start of the external clock input: (See [Figure 30.121](#))

Take the following time into account: the output AC spec of the MISO pin of this product and the input AC spec of the master reception + 1 PCLK cycle + synchronization delay.

#### (2) Continuous transmission

Write the next transmit data to TDR before the falling edge<sup>\*1</sup> of the transmit clock for bit 7. Write the transmit data to TDR in consideration of synchronization delay. If the transmit data cannot be written in time, the previous frame data is resent. (See [Figure 30.121](#))

Note 1. When CCR3.CPOL = 1 and CCR3.CPHA = 0, or CCR3.CPOL = 0 and CCR3.CPHA = 1. In the case of CCR3.CPOL = 0 and CCR3.CPHA = 0, or CCR3.CPOL = 1 and CCR3.CPHA = 1, it's the rising edge.



**Figure 30.121 Restrictions on Use of External Clock in Clock Synchronous Transmission**

### 30.19.8 Restrictions on Using DMAC or DTC

When using the DMAC or DTC to read RDR, be sure to set the receive data full interrupt (SCIn\_RXI) as the activation source of the relevant SCI.

During the operation in transmission / reception using the DMAC or DTC, it should not set transfer information of DMAC or DTC.

### 30.19.9 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IELSRn.IR flag) in the ICU is 1, follow the procedure in this section to clear interrupt requests before permitting operations (by setting the CCR0.TE or CCR0.RE bit to 1). For details on the interrupt status flag, see [section 13, Interrupt Controller Unit \(ICU\)](#).

1. Confirm that transfer has stopped (the CCR0.TE or CCR0.RE bit is 0)

2. Set the associated interrupt enable bit (CCR0.TIE or CCR0.RIE bit) to 0
3. Read the associated interrupt enable bit (CCR0.TIE or CCR0.RIE bit) to check that it actually becomes 0
4. Set the interrupt status flag, IELSRn.IR, in the ICU to 0

### 30.19.10 Limitations on Simple SPI Mode

#### (1) Master mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set in the CCR3.CPHA and CPOL bits when the CCR0.SSE bit is 1.

This prevents the clock line from being placed in the high-impedance state when the CCR0.TE bit is set to 0 or unexpected edges from being generated on the clock line when the CCR0.TE bit changes from 0 to 1. When the CCR0.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not required because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.

- For the clock delay setting (CCR3.CPHA bit is 0), the receive data full interrupt (SCIn\_RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 30.122. If the TE and RE bits in the CCR0 register become 0 before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Additionally, an SCIn\_RXI interrupt might lead to the input signal on the SSn pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master configuration, the SCKn pin output goes to high-impedance while the input on the SSn pin is at the low level if a mode fault error occurs while a character is being transferred, stopping supply of the clock signal to the connected slave. Reset the connected slave to avoid misaligned bits when transfer is restarted.

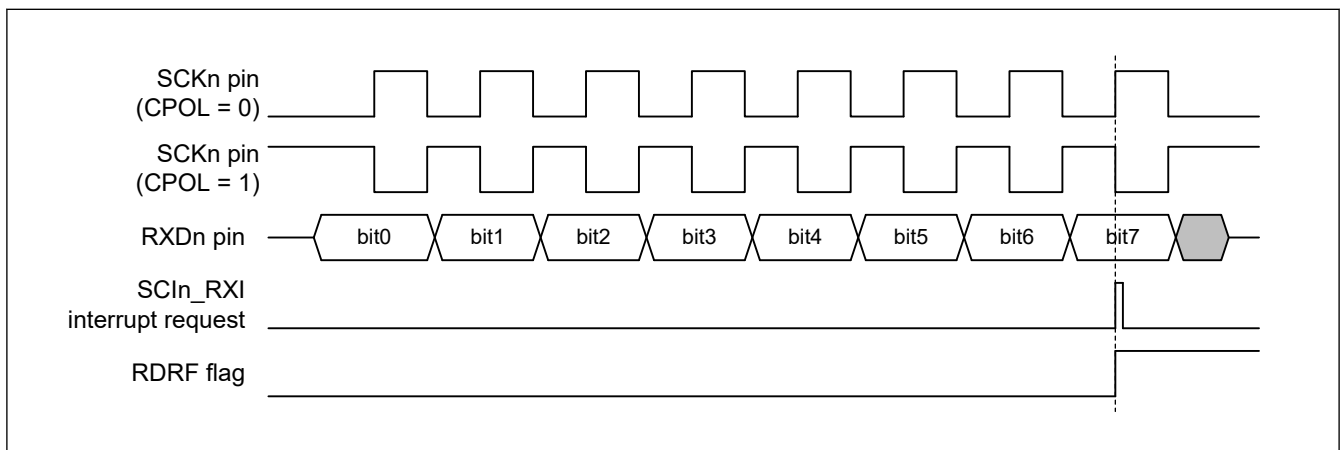


Figure 30.122 Timing of SCIn\_RXI interrupt in simple SPI mode with clock delay

#### (2) Slave mode

- It takes  $1\text{PCLK} + \text{synchronization delay time} + \text{data output delay time (AC spec)}$  from writing the transmit data to the TDR register until the data is output to the RXDn pin. Take these into account when starting external clock input.
- Provide an external clock signal to the master the same as the data length for transfer
- Secure the SSn input setup time (AC spec) from the SSn low-level input to the start of external clock input.
- Control the input on the SSn pin before the start and after the end of data transfer
- When the input level on the SSn pin is to be changed from low to high while a character is being transferred, set the TE and RE bits in the CCR0 register to 0 and, after restoring the settings, restart transfer of the first byte

### 30.19.11 Notes on Transmit Enable bit (CCR0.TE)

In initial register value, when CCR0.TE = 0, the state of the TXDn pin is high impedance. The TXDn line should not be high impedance by the following one of ways.

1. The pull-up or pull-down resistance is connected to the TXDn line.

2. Set CCR1 and decided level of TXDn terminal during TE is 0.

### 30.19.12 Notes on Simple LIN mode

In Simple LIN mode (CCR3.MOD[2:0] = 110), the following functions cannot be used.

- Multi-processor communication function
- Bit Rate Modulation function
- Loopback function
- FIFO buffer

### 30.19.13 Notes on RS-485 Driver Control function

RS-485 Driver control function is valid only in Asynchronous mode.

When RS-485 Driver control function is active (CCR3.DEN = 1), the CSR.TEND set timing / SCIn\_TEI output timing changes as follows. Wait for the SCIn\_TEI interrupt and set the CCR0.TE bit in SCI to 0.

When RS-485 Driver control function is inactive: When STOP bit output is completed.

When RS-485 Driver control function is active: At the end of DEN negation time.

### 30.19.14 Notes on Loopback function

The Loopback function is valid in Asynchronous mode with internal clock, in Manchester mode with internal clock and Clock synchronous mode with internal clock.

### 30.19.15 Notes regarding register access when operation clock (TCLK) is slower than bus clock (PCLK)

If the operating clock (TCLK) is slower than the bus clock (PCLK), the time until this information is transmitted internally after writing to the CCR0.TE and CCR0.RE registers is slower than the bus access time. In particular, when trying to change the setting register after writing 0 and interrupting communication, do not change the setting register before the signal inside the SCI is in the communication stopped state. To prevent this, after setting CCR0.TE and CCR0.RE to 0, check the CESR.TIST and CESR.RIST bits until they are 0 before setting the next register.

### 30.19.16 Notes on interrupting operation

If 0 is written to CCR0.RE during data reception and the reception operation is interrupted, there is a possibility of an invalid state, do not use the received data (RDR register stored value) and the flag value of each status register. To interrupt the reception operation, stop the interrupt or event link reception side and then write 0 to the CCR0.RE bit.

### 30.19.17 Notes on CCR3.BPEN bit setting

Set the BPEN bit only once when setting the CCR3 register in the SCI initialization flow.

This bit cannot be changed after the initialization.

When this bit setting is changed, start from the SCI initialization flow again.

## 31. I<sup>2</sup>C Bus Interface (IIC)

### 31.1 Overview

The I<sup>2</sup>C bus interface (IIC) has 2 channels. The IIC module conforms with and provides a subset of the NXP I<sup>2</sup>C (Inter-Integrated Circuit) bus interface functions.

Table 31.1 lists the IIC specifications, Figure 31.1 shows a block diagram, and Figure 31.2 shows an example of I/O pin connections to external circuits, with an I<sup>2</sup>C bus configuration. Table 31.2 lists the I/O pins.

**Table 31.1 IIC specifications (1 of 2)**

Parameter	Specifications
Communications format	<ul style="list-style-type: none"> <li>I<sup>2</sup>C-bus format or SMBus format</li> <li>Master or slave mode selectable</li> <li>Automatic securing of the setup times, hold times, and bus-free times for the transfer rate</li> </ul>
Transfer rate	<ul style="list-style-type: none"> <li>Fast-mode Plus supported, up to 1 Mbps</li> </ul>
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4% to 96%
Issuing and detecting conditions	<ul style="list-style-type: none"> <li>Start, restart, and stop conditions are automatically generated</li> <li>Start conditions (including restart conditions) and stop conditions are detectable</li> </ul>
Slave address	<ul style="list-style-type: none"> <li>Configurable for up to three different slave addresses</li> <li>7- and 10-bit address formats supported, including simultaneous use</li> <li>General call addresses, device ID addresses, and SMBus host addresses detectable</li> </ul>
Acknowledgment	<ul style="list-style-type: none"> <li>For transmission, automatic loading of the acknowledge bit Transfer of the next transmit data can be automatically suspended on detection of a not-acknowledge bit.</li> <li>For reception, automatic transmission of the acknowledge bit If a wait between the 8th and 9th clock cycles is selected, the software can control the value in the acknowledge field in response to the received value.</li> </ul>
Wait function	During reception, the following wait periods are available by holding the SCL clock low: <ul style="list-style-type: none"> <li>Waiting between the eighth and ninth clock cycles</li> <li>Waiting between the ninth clock cycle and the 1st clock cycle of the next transfer</li> </ul>
SDA output delay function	Output timing of transmitted data, including the acknowledge bit, can be delayed
Arbitration	<ul style="list-style-type: none"> <li>For multi-master operation: <ul style="list-style-type: none"> <li>SCL clock synchronization is possible when conflict occurs with the SCL signal from another master</li> <li>When issuing the start condition creates conflict on the bus, loss of arbitration is detected by testing for a mismatch between the internal signal for the SDA line and the level on the SDA line</li> <li>In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line</li> </ul> </li> <li>Loss of arbitration because the start condition occurs while the bus is busy is detectable, to prevent the issuing of double start conditions</li> <li>Loss of arbitration is detectable on transfer of a not-acknowledge bit because the internal signal for the SDA line and the level on the SDA line do not match</li> <li>Loss of arbitration because a mismatch of internal and line levels for data is detectable in slave transmission</li> </ul>
Timeout function	Internal detection of long-interval stops of the SCL clock
Noise cancellation	<ul style="list-style-type: none"> <li>Digital noise filters for both the SCL and SDA signals</li> <li>Programmable window for noise cancellation by the filters</li> </ul>
Interrupt sources	<ul style="list-style-type: none"> <li>Transfer error or event occurrence (arbitration-lost, NACK, timeout, start or restart condition, or stop condition)</li> <li>Receive data full, including matching with a slave address</li> <li>Transmit data empty, including matching with a slave address</li> <li>Transmit end</li> </ul>
Module-stop function	Module-stop state can be set for each channel to reduce power consumption.
IIC operating modes	<ul style="list-style-type: none"> <li>Master transmit</li> <li>Master receive</li> <li>Slave transmit</li> <li>Slave receive</li> </ul>

Table 31.1 IIC specifications (2 of 2)

Parameter	Specifications
Event link function (output)	<ul style="list-style-type: none"> <li>Transfer error or event occurrence (arbitration-lost, NACK, timeout, start or restart condition, or stop condition)</li> <li>Receive data full, including matching with a slave address</li> <li>Transmit data empty, including matching with a slave address</li> <li>Transmit end</li> </ul>
Wakeup function*1	CPU can return from Software Standby mode using a wakeup event
TrustZone Filter	Security and Privilege attribution can be set for each channels

Note 1. This function is only available for IIC channel IIC0. IIC1 is not supported.

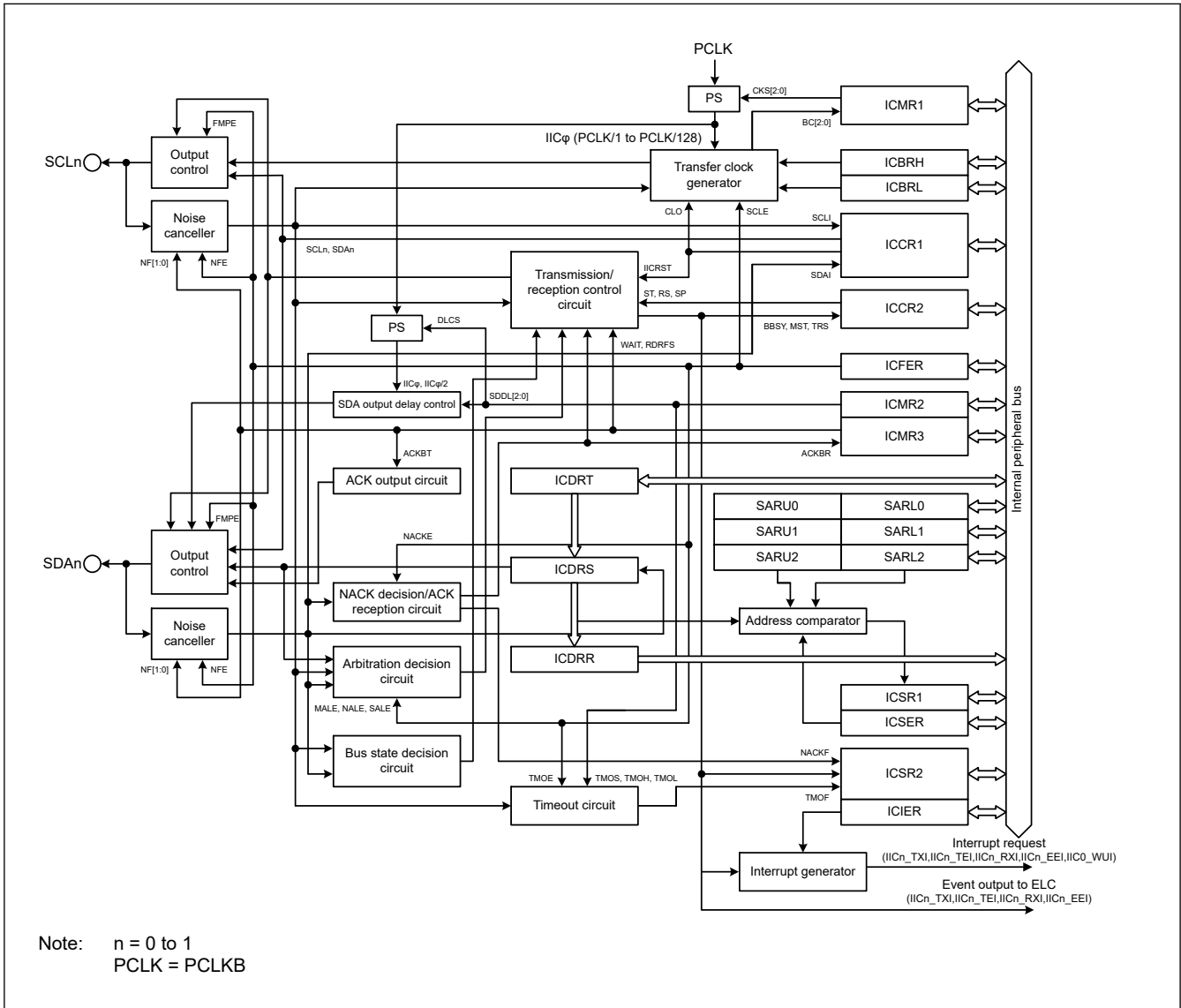
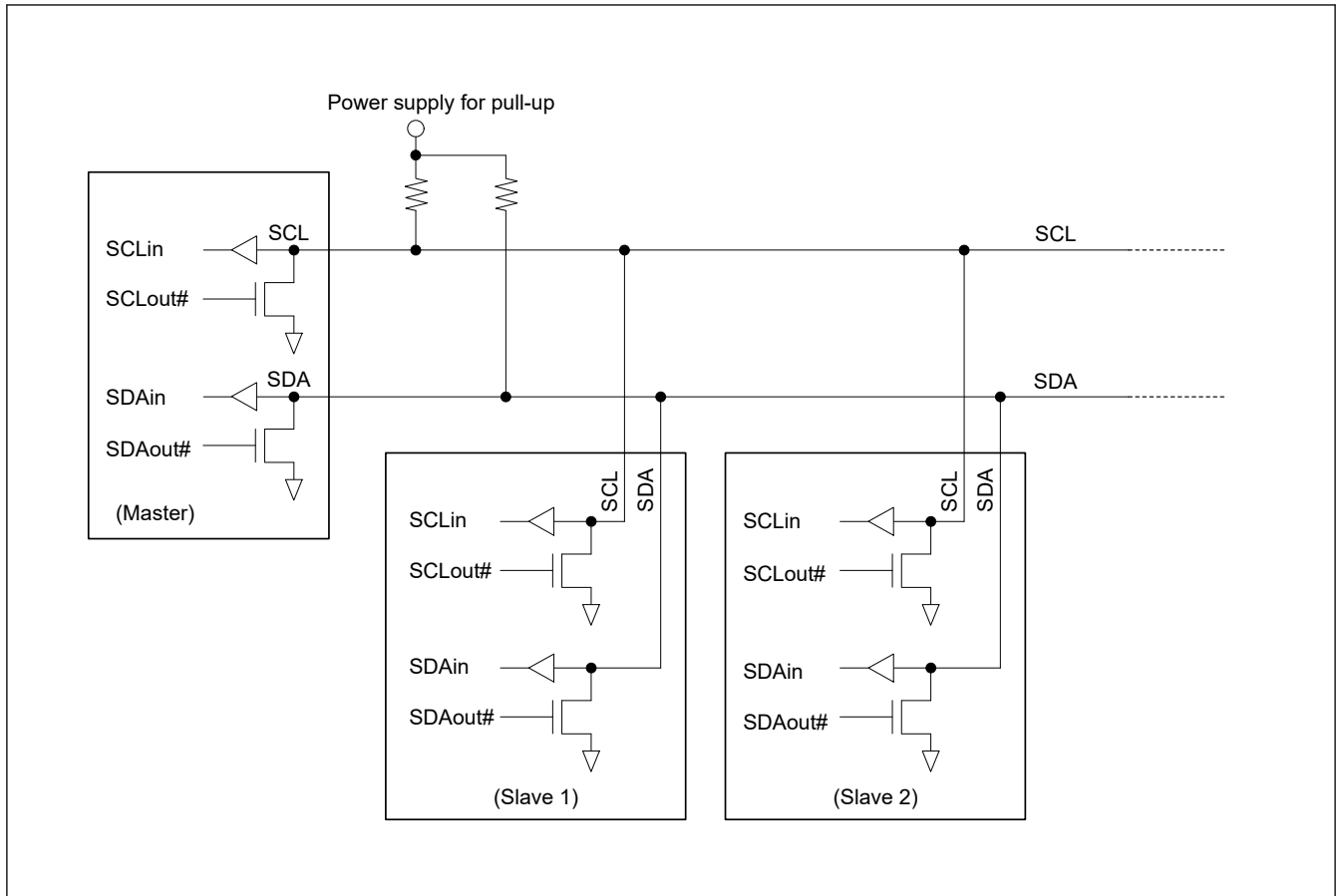


Figure 31.1 IIC block diagram





**Figure 31.2 I/O pin connection to an external circuit (I<sup>2</sup>C bus configuration example)**

The input level of the signals for IIC is CMOS when I<sup>2</sup>C bus is selected (ICMR3.SMBS = 0), or TTL when SMBus is selected (ICMR3.SMBS = 1).

**Table 31.2 IIC I/O pins**

Channel	Pin name	I/O	Function
IICn	SCLn	I/O	IICn serial clock I/O pin
	SDAn	I/O	IICn serial data I/O pin

Note: n = 0 to 1

## 31.2 Register Descriptions

### 31.2.1 ICCR1 : I<sup>2</sup>C Bus Control Register 1

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
 IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ICE	IICRS T	CLO	SOWP	SCLO	SDAO	SCLI	SDAI

Value after reset: 0 0 0 1 1 1 1 1

Bit	Symbol	Function	R/W
0	SDAI	SDA Line Monitor 0: SDAn line is low 1: SDAn line is high	R

Bit	Symbol	Function	R/W
1	SCLI	SCL Line Monitor 0: SCLn line is low 1: SCLn line is high	R
2	SDAO	SDA Output Control/Monitor 0: Read: IIC drives SDA <sub>n</sub> pin low Write: IIC drives SDA <sub>n</sub> pin low 1: Read: IIC releases SDA <sub>n</sub> pin Write: IIC releases SDA <sub>n</sub> pin	R/W
3	SCLO	SCL Output Control/Monitor Use an external pull-up resistor to drive the signal high. 0: Read: IIC drives SCL <sub>n</sub> pin low Write: IIC drives SCL <sub>n</sub> pin low 1: Read: IIC releases SCL <sub>n</sub> pin Write: IIC releases SCL <sub>n</sub> pin	R/W
4	SOWP	SCLO/SDAO Write Protect This bit is read as 1. 0: Write enable SCLO and SDAO bits 1: Write protect SCLO and SDAO bits	W
5	CLO	Extra SCL Clock Cycle Output This bit clears automatically after 1 clock cycle is output. 0: Do not output extra SCL clock cycle (default) 1: Output extra SCL clock cycle	R/W
6	IICRST	I <sup>2</sup> C Bus Interface Internal Reset This setting clears the bit counter and the SCL <sub>n</sub> /SDA <sub>n</sub> output latch. 0: Release IIC reset or internal reset 1: Initiate IIC reset or internal reset	R/W
7	ICE	I <sup>2</sup> C Bus Interface Enable Used in combination with the IICRST bit to select either IIC or internal reset. 0: Disable (SCL <sub>n</sub> and SDA <sub>n</sub> pins in inactive state) 1: Enable (SCL <sub>n</sub> and SDA <sub>n</sub> pins in active state)	R/W

Note: S-TYPE3, P-TYPE3

### SDAO bit (SDA Output Control/Monitor) and SCLO bit (SCL Output Control/Monitor)

The SDAO and SCLO bits directly control the SDA<sub>n</sub> and SCL<sub>n</sub> signals output from the IIC. When writing to these bits, also write 0 to the SOWP bit. Setting these bits results in input to the IIC by the input buffer. When slave mode is selected, a start condition might be detected and the bus might be released, depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, transmission, or reception. Operation after rewriting under these conditions is not guaranteed. When reading these bits, the state of signals output from the IIC can be read.

### CLO bit (Extra SCL Clock Cycle Output)

The CLO bit allows output of an extra SCL clock cycle for debugging or error processing. Normally, set this bit to 0. Setting the bit to 1 in a normal communication state causes a communication error. For details on this function, see [section 31.12.2. Extra SCL Clock Cycle Output Function](#).

### IICRST bit (I<sup>2</sup>C Bus Interface Internal Reset)

The IICRST bit initiates an internal state reset of the IIC. Setting this bit to 1 initiates an IIC reset or internal reset. Whether an IIC reset or internal reset is initiated is determined by the settings of this bit in combination with the ICE bit. [Table 31.3](#) lists the IIC resets.

The IIC reset initializes all registers except ICCR1.ICE and ICCR1.IICRST bits and internal states of the IIC. In addition to the internal states of the IIC, the internal reset initializes the following:

- Bit counter (ICMR1.BC[2:0] bits)
- I<sup>2</sup>C Bus Shift Register (ICDRS)
- I<sup>2</sup>C Bus Status Registers (ICSR1 and ICSR2)
- SDAO and SCLO Output Control/Monitor (ICCR1.SDAO and ICCR1.SCLO bits)

- I<sup>2</sup>C Bus Control Register 2 (except ICCR2.BBSY bit)

For the reset conditions for each register, see [section 31.15. State of Registers When Issuing Each Condition](#).

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the IIC without initializing the port settings and the control and setting registers of the IIC. If the IIC hangs in a low-level output state, resetting the internal states cancels the low-level output state and releases the bus with the SCLn pin and SDAn pin at high impedance.

**Note:** If an internal reset is initiated using the IICRST bit for a bus hang-up that occurs during communication with the master device in slave mode, the slave and master devices might enter different states, because the bit counter information differs. For this reason, do not initiate an internal reset in slave mode. Initiate recovery processing from the master device. If an internal reset is required because the IIC hangs with the SCLn line in a low-level output state in slave mode, initiate an internal reset, then issue a restart condition from the master device, or issue a stop condition and resume communication from the start condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start or restart condition from the master device, synchronization is lost because the master and slave devices operate asynchronously.

**Table 31.3 IIC resets**

IICRST	ICE	State	Specifications
1	0	IIC reset	Resets all registers except ICCR1.IICRST and ICCR1.ICE bits, and the internal states of the IIC
	1	Internal reset	Reset the following: <ul style="list-style-type: none"> <li>• ICMR1.BC[2:0] bits</li> <li>• ICSR1, ICSR2, ICDRS registers</li> <li>• SDAO and SCLO Output Control/Monitor (ICCR1.SDAO and ICCR1.SCLO bits)</li> <li>• I<sup>2</sup>C Bus Control Register 2 (except ICCR2.BBSY bit)</li> <li>• Internal states of the IIC</li> </ul>

### ICE bit (I<sup>2</sup>C Bus Interface Enable)

The ICE bit selects the active or inactive state of the SCLn and SDAn pins. It can also be combined with the IICRST bit to initiate two types of resets. See [Table 31.3](#) for the reset descriptions.

Set the ICE bit to 1 when using the IIC. The SCLn and SDAn pins are placed in the active state when the ICE bit is set to 1. Set the ICE bit to 0 when the IIC is not used. The SCLn and SDAn pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCLn or SDAn pin to the IIC when setting up the pin function control. Slave address comparison is performed if the pins are assigned to the IIC.

### 31.2.2 ICCR2 : I<sup>2</sup>C Bus Control Register 2

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ST	Start Condition Issuance Request 0: Do not issue a start condition request 1: Issue a start condition request	R/W
2	RS	Restart Condition Issuance Request 0: Do not issue a restart condition request 1: Issue a restart condition request	R/W

Bit	Symbol	Function	R/W
3	SP	Stop Condition Issuance Request 0: Do not issue a stop condition request 1: Issue a stop condition request	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	TRS	Transmit/Receive Mode 0: Receive mode 1: Transmit mode	R/W <sup>1</sup>
6	MST	Master/Slave Mode 0: Slave mode 1: Master mode	R/W <sup>1</sup>
7	BBSY	Bus Busy Detection Flag 0: I <sup>2</sup> C bus released (bus free state) 1: I <sup>2</sup> C bus occupied (bus busy state)	R

Note: S-TYPE3, P-TYPE3

Note 1. The MST and TRS bits can be written to when the ICMR1.MTWP bit is set to 1.

### ST bit (Start Condition Issuance Request)

The ST bit requests transition to master mode and triggers a start condition. When this bit is set to 1, a start condition is issued when the BBSY flag is set to 0 (bus free state). For details on this function, see [section 31.11. Start, Restart, and Stop Condition Issuing Function](#).

[Setting condition]

- When 1 is written to the ST bit.

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Only set the ST bit to 1 (start condition request) when the BBSY flag is set to 0 (bus free state). Arbitration might be lost if the ST bit is set to 1 (start condition request) when the BBSY flag is 1 (bus busy state).

### RS bit (Restart Condition Issuance Request)

The RS bit requests that a restart condition be issued in master mode. When this bit is set to 1 to request a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see [section 31.11. Start, Restart, and Stop Condition Issuing Function](#).

[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition is issued (a start condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Do not set the RS bit to 1 while issuing a stop condition.

Note: If 1 (restart condition request) is written to the RS bit in slave mode, the restart condition is not issued, but the RS bit remains set to 1. If the operating mode changes to master mode without the bit being cleared, a restart condition might be issued.

**SP bit (Stop Condition Issuance Request)**

The SP bit requests that a stop condition be issued in master mode. When this bit is set to 1, a stop condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see [section 31.11. Start, Restart, and Stop Condition Issuing Function](#).

[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the MST bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition is issued (a stop condition is detected)
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Writing to the SP bit is not possible while the BBSY flag is 0 (bus free state).

Note: Do not set the SP bit to 1 while a restart condition is being issued.

**TRS bit (Transmit/Receive Mode)**

The TRS bit indicates transmit or receive mode. The IIC is in receive mode when the TRS bit is 0 and in transmit mode when the bit is 1. The combination of this bit and the MST bit indicates the IIC operating mode.

The value of the TRS bit automatically changes to 1 for transmit mode or 0 for receive mode when a start condition is issued or detected and the R/W# bit is set. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is issued normally because of a restart condition request (when a restart condition is detected with the RS bit set to 1)
- When the R/W# bit appended to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in ICSESR, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When the R/W# bit appended to the slave address is set to 1 in master mode
- In slave mode, on a match between the received address and the address enabled in ICSESR when the value of the received R/W# bit is 0, including when the received address is the general call address
- In slave mode, when a restart condition is detected (a restart condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0)
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

**MST bit (Master/Slave Mode)**

The MST bit indicates master or slave mode. The IIC is in slave mode when the MST bit is 0 and is in master mode when the bit is 1. The combination of this bit and the TRS bit indicates the IIC operating mode.

The value of the MST bit automatically changes to 1 for master mode or 0 for slave mode when a start condition is issued or a stop condition is issued or detected. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

## [Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1.

## [Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

**BBSY flag (Bus Busy Detection Flag)**

The BBSY flag indicates whether the I<sup>2</sup>C bus is occupied (bus busy state) or released (bus free state). The flag is set to 1 when the SDAn line changes from high to low when the SCLn line is high, assuming that a start condition was issued. The flag is set to 0 if a start condition is not detected for the bus free time (ICBRL setting), assuming that a stop condition was issued.

## [Setting condition]

- When a start condition is detected.

## [Clearing conditions]

- When a start condition is not detected for the bus free time (ICBRL setting) after detecting a stop condition
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (IIC reset).

**31.2.3 ICMR1 : I<sup>2</sup>C Bus Mode Register 1**

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MTWP	CKS[2:0]		BCWP	BC[2:0]			
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	BC[2:0]	Bit Counter 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W <sup>*1</sup>
3	BCWP	BC Write Protect This bit is read as 1. 0: Write enable BC[2:0] bits 1: Write protect BC[2:0] bits	W <sup>*1</sup>
6:4	CKS[2:0]	Internal Reference Clock Select Select the internal reference clock source (IICφ) for the IIC. IICφ = (PCLKB / 2 <sup>CKS[2:0]</sup> ) clock	R/W
7	MTWP	MST/TRS Write Protect 0: Write protect MST and TRS bits in ICCR2 1: Write enable MST and TRS bits in ICCR2	R/W

Note: S-TYPE3, P-TYPE3

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

### BC[2:0] bits (Bit Counter)

The BC[2:0] bits function as a counter indicating the number of bits remaining to be transferred on detection of a rising edge on the SCLn line. Although BC[2:0] are read/write bits, it is not required to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one, for an additional acknowledge bit, between transferred frames when the SCLn line is at a low level. The value in the BC[2:0] bits returns to 000b at the end of a data transfer, including the acknowledge bit, or when a start or restart condition is detected.

## 31.2.4 ICMR2 : I<sup>2</sup>C Bus Mode Register 2

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
0	TMOS	Timeout Detection Time Select 0: Select long mode 1: Select short mode	R/W
1	TMOL	Timeout L Count Control 0: Disable count while SCLn line is low 1: Enable count while SCLn line is low	R/W
2	TMOH	Timeout H Count Control 0: Disable count while SCLn line is high 1: Enable count while SCLn line is high	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	SDDL[2:0]	SDA Output Delay Counter 0 0 0: No output delay 0 0 1: 1 IICφ cycle (When ICMR2.DLCS = 0 (IICφ)) 1 or 2 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 0 1 0: 2 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 3 or 4 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 0 1 1: 3 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 5 or 6 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 0 0: 4 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 7 or 8 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 0 1: 5 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 9 or 10 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 1 0: 6 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 11 or 12 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2)) 1 1 1: 7 IICφ cycles (When ICMR2.DLCS = 0 (IICφ)) 13 or 14 IICφ cycles (When ICMR2.DLCS = 1 (IICφ/2))	R/W
7	DLCS	SDA Output Delay Clock Source Select 0: Select internal reference clock (IICφ) as the clock source for SDA output delay counter 1: Select internal reference clock divided by 2 (IICφ/2) as the clock source for SDA output delay counter <sup>*1</sup>	R/W

Note: S-TYPE3, P-TYPE3

Note 1. The setting DLCS = 1 (IICφ/2) is only valid when SCL is low. When SCL is high, the DLCS = 1 setting becomes invalid and the clock source becomes the internal reference clock (IICφ).

### TMOS bit (Timeout Detection Time Select)

The TMOS bit selects long or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE = 1). When this bit is set to 0, long mode is selected. When it is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a

14-bit counter. While the SCLn line is in the state that enables this counter as specified in the TMOH and TMOL bits, the counter counts up in synchronization with the internal reference clock (IIC $\phi$ ) as a count source. For details on this function, see [section 31.12.1. Timeout Function](#).

### TMOL bit (Timeout L Count Control)

The TMOL bit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held low and the timeout function is enabled (ICFER.TMOE = 1).

### TMOH bit (Timeout H Count Control)

The TMOH bit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held high and the timeout function is enabled (ICFER.TMOE = 1).

### SDDL[2:0] bits (SDA Output Delay Counter)

The SDDL[2:0] bits can be used to delay the SDA output. This counter works with the clock source selected in the DLCS bit. This setting can be used for all types of SDA output, including transmission of the acknowledge bit.

Set the SDA output delay to meet the I<sup>2</sup>C bus standard for the data enable time/acknowledge enable time,<sup>\*1</sup> or the SMBus standard, within [data hold time (300 ns or more + the SCL clock low-level period) - the data setup time (250 ns)]. If a value outside the standard is set, communication between devices might malfunction or falsely indicate a start or stop condition, depending on the bus state.

For details on this function, see [section 31.5. SDA Output Delay Function](#).

Note 1. Data enable time/acknowledge enable time.

3,450 ns for up to 100 kbps: Standard mode (Sm)

900 ns for up to 400 kbps: Fast mode (Fm)

450 ns for up to 1 Mbps: Fast-mode Plus (Fm+)

## 31.2.5 ICMR3 : I<sup>2</sup>C Bus Mode Register 3

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SMBS	WAIT	RDRF S	ACKW P	ACKB T	ACKB R	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	NF[1:0]	Noise Filter Stage Select 0 0: Filter out noise of up to 1 IIC $\phi$ cycle (single-stage filter) 0 1: Filter out noise of up to 2 IIC $\phi$ cycles (2-stage filter) 1 0: Filter out noise of up to 3 IIC $\phi$ cycles (3-stage filter) 1 1: Filter out noise of up to 4 IIC $\phi$ cycles (4-stage filter)	R/W
2	ACKBR	Receive Acknowledge 0: 0 received as the acknowledge bit (ACK reception) 1: 1 received as the acknowledge bit (NACK reception)	R
3	ACKBT	Transmit Acknowledge 0: Send 0 as the acknowledge bit (ACK transmission) 1: Send 1 as the acknowledge bit (NACK transmission)	R/W <sup>*1</sup>
4	ACKWP	ACKBT Write Protect 0: Write protect ACKBT bit 1: Write enable ACKBT bit	R/W



Bit	Symbol	Function	R/W
5	RDRFS	RDRF Flag Set Timing Select Low-hold is released by writing to ACKBT. 0: Set the RDRF flag on the rising edge of the 9th SCL clock cycle. The SCLn line is not held low on the falling edge of the 8th clock cycle. 1: Set the RDRF flag on the rising edge of the 8th SCL clock cycle. The SCLn line is held low on the falling edge of the 8th clock cycle.	R/W <sup>2</sup>
6	WAIT	Low-hold is released by reading ICDRR. 0: No wait (The SCLn line is not held low during the period between the 9th clock cycle and the 1st clock cycle.) 1: Wait (The SCLn line is held low during the period between the 9th clock cycle and the 1st clock cycle.)	R/W <sup>2</sup>
7	SMBS	SMBus/I <sup>2</sup> C Bus Select 0: Select I <sup>2</sup> C Bus 1: Select SMBus	R/W

Note: S-TYPE3, P-TYPE3

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If the application writes 1 to the ACKWP and ACKBT bits at the same time, the ACKBT bit is not set to 1.

Note 2. The WAIT and RDRFS bits are only valid in receive mode (invalid in transmit mode).

### NF[1:0] bits (Noise Filter Stage Select)

The NF[1:0] bits select the number of stages in the digital noise filter. For details on this function, see [section 31.6. Digital Noise Filter Circuits](#)

Note: Set the noise range to be filtered within a range less than the SCLn line high- or low-level period. If the noise range is set to a value of [SCL clock width: high- or low-level period, whichever is shorter] - [1.5 internal reference clock (IIC $\phi$ ) cycles + analog noise filter: 120 ns (reference values)] or more, the SCL clock is regarded as noise, which might prevent the IIC from operating normally.

### ACKBR bit (Receive Acknowledge)

The ACKBR bit stores the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

### ACKBT bit (Transmit Acknowledge)

The ACKBT bit sets the acknowledge bit to be sent in receive mode

[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1.

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition request is detected with the SP bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

### ACKWP bit (ACKBT Write Protect)

The ACKWP bit controls write enabling of the ACKBT bit.

### RDRFS bit (RDRF Flag Set Timing Select)

The RDRFS bit selects the RDRF flag set timing in receive mode and also selects whether to hold the SCLn line low on the falling edge of the 8th SCL clock cycle.

When the RDRFS bit is 0, the SCLn line is not held low on the falling edge of the 8th SCL clock cycle, and the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 on the rising edge of the 8th SCL clock cycle, and the SCLn line is held low on the falling edge of the 8th SCL clock cycle. The low-hold of the SCLn line is released by a write to the ACKBT bit.

After data is received with this setting, the SCLn line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1), based on the receive data.

### WAIT bit (WAIT)

The WAIT bit controls whether to force a low-hold between the ninth SCL clock cycle and the first SCL clock cycle, until the receive data buffer (ICDRR) is completely read each time single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation continues without a low-hold between the ninth and the first SCL clock cycle. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCLn line is held low from the falling edge of the ninth clock cycle until the ICDRR value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, always read ICDRR first.

### SMBS bit (SMBus/I<sup>2</sup>C Bus Select)

Setting the SMBS bit to 1 selects the SMBus and enables the HOAE bit in ICSESR.

## 31.2.6 ICFER : I<sup>2</sup>C Bus Function Enable Register

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	FMPE	SCLE	NFE	NACK E	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Function	R/W
0	TMOE	Timeout Function Enable 0: Disable 1: Enable	R/W
1	MALE	Master Arbitration-Lost Detection Enable 0: Disable the arbitration-lost detection function and disable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost 1: Enable the arbitration-lost detection function and enable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost	R/W
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
3	SALE	Slave Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
4	NACKE	NACK Reception Transfer Suspension Enable 0: Do not suspend transfer operation during NACK reception (disable transfer suspension) 1: Suspend transfer operation during NACK reception (enable transfer suspension)	R/W
5	NFE	Digital Noise Filter Circuit Enable 0: Do not use the digital noise filter circuit 1: Use the digital noise filter circuit	R/W
6	SCLE	SCL Synchronous Circuit Enable 0: Do not use the SCL synchronous circuit 1: Use the SCL synchronous circuit	R/W

Bit	Symbol	Function	R/W
7	FMPE <sup>*1</sup>	Fast-Mode Plus Enable 0: Do not use the Fm+ slope control circuit for the SCLn and SDAn pins 1: Use the Fm+ slope control circuit for the SCLn and SDAn pins.	R/W

Note: S-TYPE3, P-TYPE3

Note 1. The Fast-mode Plus enable bit (FMPE) is supported by IIC0 (SCL0\_A, SDA0\_A) and IIC1(SCL1\_A, SDA1\_A). Bit [7] is the reserved bit in the not supported channel.

### TMOE bit (Timeout Function Enable)

The TMOE bit enables or disables the timeout function. For details on this function, see [section 31.12.1. Timeout Function](#).

### MALE bit (Master Arbitration-Lost Detection Enable)

The MALE bit specifies whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

### NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

The NALE bit specifies whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode, for example when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with a different number of receive bytes.

### SALE bit (Slave Arbitration-Lost Detection Enable)

The SALE bit specifies whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode, for example when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs because of noise.

### NACKE bit (NACK Reception Transfer Suspension Enable)

The NACKE bit specifies whether to continue or discontinue the transfer operation when NACK is received in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended. When the NACKE bit is 0, the next transfer operation continues regardless of the received acknowledge content.

For details, see [section 31.9.2. NACK Reception Transfer Suspension Function](#).

### SCLE bit (SCL Synchronous Circuit Enable)

The SCLE bit specifies whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCLE bit is set to 0 (no SCL synchronous circuit used), the IIC does not synchronize the SCL clock with the SCL input clock. With this setting, the IIC outputs the SCL clock at the transfer rate set in ICBRH and ICBRL, regardless of the SCLn line state. For this reason, if the bus load of the I<sup>2</sup>C bus line is much larger than the specification value, or if the SCL clock output overlaps in multiple masters, a short-cycle SCL clock that does not meet the specification might be output. When no SCL synchronous circuit is used, it also affects the issuing of the start, restart, and stop conditions, and the continuous output of extra SCL clock cycles.

Do not set this bit to 0 except when checking the output of the set transfer rate.

### FMPE bit (Fast-Mode Plus Enable)

The FMPE bit specifies whether to use a slope control circuit for Fast-mode Plus (Fm+).

When this bit is set to 1, a slope control circuit conforming to the I<sup>2</sup>C bus Fast-mode Plus (Fm+) standard (tof) is selected. When this bit is set to 0, a slope control circuit conforming to the I<sup>2</sup>C bus Standard-mode (Sm) and Fast-mode (Fm) standards (tof) is selected.

Set this bit to 1 when using transmission rates up to 1 Mbps (Fast-mode Plus (Fm+) standard). Set it to 0 when using other transmission rates (up to 100 kbps (Sm) or up to 400 kbps (Fm)) or for SMBus (10 to 100 kbps).

### 31.2.7 IC SER : I<sup>2</sup>C Bus Status Enable Register

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOAE	—	DIDE	—	GCAE	SAR2 E	SAR1 E	SAR0 E

Value after reset: 0 0 0 0 1 0 0 1

Bit	Symbol	Function	R/W
0	SAR0E	Slave Address Register 0 Enable 0: Disable slave address in SARL0 and SARU0 1: Enable slave address in SARL0 and SARU0	R/W
1	SAR1E	Slave Address Register 1 Enable 0: Disable slave address in SARL1 and SARU1 1: Enable slave address in SARL1 and SARU1	R/W
2	SAR2E	Slave Address Register 2 Enable 0: Disable slave address in SARL2 and SARU2 1: Enable slave address in SARL2 and SARU2	R/W
3	GCAE	General Call Address Enable 0: Disable general call address detection 1: Enable general call address detection	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DIDE	Device-ID Address Detection Enable 0: Disable device-ID address detection 1: Enable device-ID address detection	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HOAE	Host Address Enable 0: Disable host address detection 1: Enable host address detection	R/W

Note: S-TYPE3, P-TYPE3

#### SARyE bit (Slave Address Register y Enable) (y = 0 to 2)

The SARyE bit enables or disables the received slave address and the slave address set in SARLy and SARUy.

When this bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address. When this bit is set to 0, the slave address set in SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

#### GCAE bit (General Call Address Enable)

The GCAE bit specifies whether to ignore the general call address (0000 000b + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the IIC recognizes the received slave address as the general call address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the data receive operation. When this bit is set to 0, the received slave address is ignored even if it matches the general call address.

#### DIDE bit (Device-ID Address Detection Enable)

The DIDE bit specifies whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first frame after a start or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the IIC recognizes that the device-ID address was received. When the next R/W# bit is 0 (W), the IIC recognizes the second and the subsequent frames as slave addresses and continues the receive operation. When this bit is set to 0, the IIC ignores the received first frame even if it matches the device-ID address, and it recognizes the first frame as a normal slave address.

For details on this function, see [section 31.7.3. Device-ID Address Detection](#).

**HOAE bit (Host Address Enable)**

The HOAE bit specifies whether to ignore the received host address (0001 000b) when the SMBS bit in ICMR3 is 1.

When this bit is set to 1 while the SMBS bit in ICMR3 is 1, if the received slave address matches the host address, the IIC recognizes the received slave address as the host address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the receive operation.

When the SMBS bit in ICMR3 or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

**31.2.8 ICIER : I<sup>2</sup>C Bus Interrupt Enable Register**

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOIE	Timeout Interrupt Request Enable 0: Disable timeout interrupt (TMOI) request 1: Enable timeout interrupt (TMOI) request	R/W
1	ALIE	Arbitration-Lost Interrupt Request Enable 0: Disable arbitration-lost interrupt (ALI) request 1: Enable arbitration-lost interrupt (ALI) request	R/W
2	STIE	Start Condition Detection Interrupt Request Enable 0: Disable start condition detection interrupt (STI) request 1: Enable start condition detection interrupt (STI) request	R/W
3	SPIE	Stop Condition Detection Interrupt Request Enable 0: Disable stop condition detection interrupt (SPI) request 1: Enable stop condition detection interrupt (SPI) request	R/W
4	NAKIE	NACK Reception Interrupt Request Enable 0: Disable NACK reception interrupt (NAKI) request 1: Enable NACK reception interrupt (NAKI) request	R/W
5	RIE	Receive Data Full Interrupt Request Enable 0: Disable receive data full interrupt (IICn_RXI) request 1: Enable receive data full interrupt (IICn_RXI) request	R/W
6	TEIE	Transmit End Interrupt Request Enable 0: Disable transmit end interrupt (IICn_TEI) request 1: Enable transmit end interrupt (IICn_TEI) request	R/W
7	TIE	Transmit Data Empty Interrupt Request Enable 0: Disable transmit data empty interrupt (IICn_TXI) request 1: Enable transmit data empty interrupt (IICn_TXI) request	R/W

Note: S-TYPE3, P-TYPE3

**TMOIE bit (Timeout Interrupt Request Enable)**

The TMOIE bit enables or disables timeout interrupt (TMOI) requests when the TMOF flag in ICSR2 is 1. To cancel a TMOI interrupt request, set the TMOF flag or the TMOIE bit to 0.

**ALIE bit (Arbitration-Lost Interrupt Request Enable)**

The ALIE bit enables or disables arbitration-lost interrupt (ALI) requests when the AL flag in ICSR2 is 1. To cancel an ALI interrupt request, set the AL flag or the ALIE bit to 0.

**STIE bit (Start Condition Detection Interrupt Request Enable)**

The STIE bit enables or disables start condition detection interrupt (STI) requests when the START flag in ICSR2 is 1. To cancel an STI interrupt request, set the START flag or the STIE bit to 0.

**SPIE bit (Stop Condition Detection Interrupt Request Enable)**

The SPIE bit enables or disables stop condition detection interrupt (SPI) requests when the STOP flag in ICSR2 is 1. To cancel an SPI interrupt request, set the STOP flag or the SPIE bit to 0.

**NAKIE bit (NACK Reception Interrupt Request Enable)**

The NAKIE bit enables or disables NACK reception interrupt (NAKI) requests when the NACKF flag in ICSR2 is 1. To cancel an NAKI interrupt request, set the NACKF flag or the NAKIE bit to 0.

**RIE bit (Receive Data Full Interrupt Request Enable)**

The RIE bit enables or disables receive data full interrupt (IICn\_RXI) requests when the RDRF flag in ICSR2 is 1.

**TEIE bit (Transmit End Interrupt Request Enable)**

The TEIE bit enables or disables transmit end interrupt (IICn\_TEI) requests when the TEND flag in ICSR2 is 1. To cancel an IICn\_TEI interrupt request, set the TEND flag or the TEIE bit to 0.

**TIE bit (Transmit Data Empty Interrupt Request Enable)**

The TIE bit enables or disables transmit data empty interrupt (IICn\_TXI) requests when the TDRE flag in ICSR2 is 1.

**31.2.9 ICSR1 : I<sup>2</sup>C Bus Status Register 1**

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 not detected 1: Slave address 0 detected	R/(W) <sup>*1</sup>
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 not detected 1: Slave address 1 detected	R/(W) <sup>*1</sup>
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 not detected 1: Slave address 2 detected	R/(W) <sup>*1</sup>
3	GCA	General Call Address Detection Flag 0: General call address not detected 1: General call address detected	R/(W) <sup>*1</sup>
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DID	Device-ID Address Detection Flag This bit is set to 1 when the first frame received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0[W]). 0: Device-ID command not detected 1: Device-ID command detected	R/(W) <sup>*1</sup>
6	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	HOA	Host Address Detection Flag This bit is set to 1 when the received slave address matches the host address (0001 000b). 0: Host address not detected 1: Host address detected	R/(W) <sup>*1</sup>

Note: S-TYPE3, P-TYPE3

Note 1. Only 0 can be written to clear the flag.

### AASy flag (Slave Address y Detection flag) (y = 0 to 2)

The AASy flag indicates whether slave address y was detected.

[Setting conditions]

For 7-bit address format (SARUy.FS = 0):

- When the received slave address matches the SVA[6:0] value in SARLy, with the SARyE bit in ICSEr set to 1 (slave address y detection enabled).  
The AASy flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format (SARUy.FS = 1):

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address matches the SARLy value, with the SARyE bit in ICSEr set to 1 (slave address y detection enabled).  
The AASy flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy flag after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

For 7-bit address format (SARUy.FS = 0):

- When the received slave address does not match the SVA[6:0] value in SARLy, with the SARyE bit in ICSEr set to 1 (slave address y detection enabled).  
The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

For 10-bit address format (SARUy.FS = 1):

- When the received slave address does not match a value of (11110b + SVA[1:0] in SARUy), with the SARyE bit in ICSEr set to 1 (slave address y detection enabled).  
The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address does not match the SARLy value, with the SARyE bit in ICSEr set to 1 (slave address y detection enabled).  
The AASy flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

### GCA flag (General Call Address Detection Flag)

The GCA flag indicates whether the general call address was detected.

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSEr set to 1 (general call address detection enabled).  
The GCA flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSEr set to 1 (general call address detection enabled).  
The GCA flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.

- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

### DID flag (Device-ID Address Detection Flag)

The DID flag indicates whether the device-ID address was detected.

[Setting condition]

- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), with the DIDE bit in IC SER set to 1 (device-ID address detection enabled).  
The DID flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start or restart condition is detected does not match a value of (device ID (1111 100b)), with the DIDE bit in IC SER set to 1 (device-ID address detection enabled)  
The DID flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), and the second frame does not match any slave address from 0 to 2, with the DIDE bit in IC SER set to 1 (device-ID address detection enabled)  
The DID flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

### HOA flag (Host Address Detection Flag)

The HOA flag indicates whether the host address was detected.

[Setting condition]

- When the received slave address matches the host address (0001 000b), with the HOAE bit in IC SER set to 1 (host address detection enabled).  
The HOA flag is set to 1 on the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA = 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b), with the HOAE bit in IC SER set to 1 (host address detection enabled)  
The HOA flag is set to 0 on the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

## 31.2.10 ICSR2 : I<sup>2</sup>C Bus Status Register 2

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDRE	TEND	RDRF	NACK F	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOF	Timeout Detection Flag 0: Timeout not detected 1: Timeout detected	R/(W) <sup>1</sup>



Bit	Symbol	Function	R/W
1	AL	Arbitration-Lost Flag 0: Arbitration not lost 1: Arbitration lost	R/(W) <sup>*1</sup>
2	START	Start Condition Detection Flag 0: Start condition not detected 1: Start condition detected	R/(W) <sup>*1</sup>
3	STOP	Stop Condition Detection Flag 0: Stop condition not detected 1: Stop condition detected	R/(W) <sup>*1</sup>
4	NACKF	NACK Detection Flag 0: NACK not detected 1: NACK detected	R/(W) <sup>*1</sup>
5	RDRF	Receive Data Full Flag 0: ICDRR contains no receive data 1: ICDRR contains receive data	R/(W) <sup>*1</sup>
6	TEND	Transmit End Flag 0: Data being transmitted 1: Data transmit complete	R/(W) <sup>*1</sup>
7	TDRE	Transmit Data Empty Flag 0: ICDRT contains transmit data 1: ICDRT contains no transmit data	R

Note: S-TYPE3, P-TYPE3

Note 1. Only 0 can be written, to clear the flag.

### TMOF flag (Timeout Detection Flag)

The TMOF flag is set to 1 when the IIC detects a timeout because the SCLn line state remains unchanged for the set period.

[Setting condition]

- When the SCLn line state remains unchanged for the period specified in the ICMR2.TMOH, TMOL, and TMOS bits while the ICFER.TMOE bit is 1 (timeout function enabled) in master or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF flag after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

### AL flag (Arbitration-Lost Flag)

The AL flag indicates that bus mastership was lost in arbitration because of a bus conflict or some other reason when a start condition was issued or an address and data was transmitted. The IIC monitors the level on the SDA<sub>n</sub> line during transmission and, if the level on the line does not match the value of the bit being output, is set the value of the AL flag to 1 to indicate that the bus is occupied by another device.

The IIC can also set the AL flag to indicate the detection of arbitration loss during NACK transmission or during data transmission.

[Setting conditions]

When master arbitration-lost detection is enabled (ICFER.MALE = 1):

- When the internal SDA output state does not match the SDA<sub>n</sub> line level on the rising edge of the SCL clock except for the ACK period during data transmission in master transmit mode
- When a start condition is detected while the ST bit in ICCR2 is 1 (start condition requested) or the internal SDA output state does not match the SDA<sub>n</sub> line level
- When the ST bit in ICCR2 is 1 (start condition requested), with the BBSY flag in ICCR2 set to 1.

When NACK arbitration-lost detection is enabled (ICFER.NALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled (ICFER.SALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock, except for the ACK period during data transmission in slave transmit mode.

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

**Table 31.4 Relationship between arbitration-lost generation sources and arbitration-lost enable functions**

ICFER			ICSR2	Error	Arbitration-lost generation source
MALE	NALE	SALE	AL		
1	x	x	1	Start condition issuance error	When internal SDA output state does not match SDAn line level when a start condition is detected, while the ST bit in ICCR2 is 1
					When ST in ICCR2 is set to 1 while BBSY in ICCR2 is 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
x	1	x	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master or slave receive mode
x	x	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

x: Don't care

### START flag (Start Condition Detection Flag)

The START flag indicates whether a start or restart condition was detected.

[Setting condition]

- When a start or restart condition is detected.

[Clearing conditions]

- When 0 is written to the START flag after reading START = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

### STOP flag (Stop Condition Detection Flag)

The STOP flag indicates whether a stop condition was detected.

[Setting condition]

- When a stop condition is detected.

[Clearing conditions]

- When 0 is written to the STOP flag after reading STOP = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

### NACKF flag (NACK Detection Flag)

The NACKF flag indicates whether a NACK was detected.

[Setting condition]

- When acknowledge is not received (NACK received) from the receive device in transmit mode, with the NACKE bit in ICFER set to 1 (transfer suspension enabled).

## [Clearing conditions]

- When 0 is written to the NACKF flag after reading NACKF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1, the IIC suspends data transmission and reception. Writing to ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit or receive operation. To restart data transmission or reception, set the NACKF flag to 0.

**RDRF flag (Receive Data Full Flag)**

The RDRF flag indicates whether the ICDRR contains receive data.

## [Setting conditions]

- When receive data is transferred from ICDRS to ICDRR  
The RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL clock cycle (selected in the RDRFS bit in ICMR3).
- When the received slave address matches after a start or restart condition is detected with the TRS bit in ICCR2 set to 0.

## [Clearing conditions]

- When 0 is written to the RDRF flag after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

**TEND flag (Transmit End Flag)**

The TEND flag indicates completion of transmission.

## [Setting condition]

- On the rising edge of the ninth SCL clock cycle while the TDRE flag is 1.

## [Clearing conditions]

- When 0 is written to the TEND flag after reading TEND = 1
- When data is written to ICDRT
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

**TDRE flag (Transmit Data Empty Flag)**

The TDRE flag indicates whether the ICDRT contains transmit data.

## [Setting conditions]

- When data is transferred from ICDRT to ICDRS and ICDRT becomes empty
- When the TRS bit in ICCR2 is set to 1
- When the received slave address matches while the TRS bit is 1.

## [Clearing conditions]

- When data is written to ICDRT
- When the TRS bit in ICCR2 is set to 0
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1 while the NACKE bit in ICFER is 1, the IIC suspends data transmission and reception. In this case, if the TDRE flag is 0 (next transmit data written), data is transferred to the ICDRS register and the ICDRT register becomes empty on the rising edge of the 9th clock cycle, but the TDRE flag does not set to 1.

### 31.2.11 ICWUR : I<sup>2</sup>C Bus Wakeup Unit Register

Base address: IIC0WU = 0x4025\_E014  
IIC0WU\_NS = 0x5025\_E014

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WUE	WUIE	WUF	WUACK	—	—	—	WUAF A

Value after reset: 0 0 0 1 0 0 0 0

Bit	Symbol	Function	R/W
0	WUAF A	Wakeup Analog Filter Additional Selection 0: Do not add the wakeup analog filter 1: Add the wakeup analog filter	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	WUACK	ACK Bit for Wakeup Mode Choice of four response modes in combination with ICCR1.IICRST and WUACK. See <a href="#">Table 31.5</a> .	R/W
5	WUF	Wakeup Event Occurrence Flag 0: Slave address not matching during wakeup 1: Slave address matching during wakeup	R/W
6	WUIE	Wakeup Interrupt Request Enable 0: Disable wakeup interrupt request (IIC0_WUI) 1: Enable wakeup interrupt request (IIC0_WUI)	R/W
7	WUE	Wakeup Function Enable 0: Disable wakeup function 1: Enable wakeup function	R/W

Note: S-TYPE3, P-TYPE3

**Table 31.5 Wakeup mode**

IICRST	WUACK	Operation mode	Description
0	0	Normal wakeup mode 1	ACK response on 9th SCL, and SCL low-hold after 9th SCL.
0	1	Normal wakeup mode 2	No ACK response immediately and SCL low-hold between 8th and 9th SCL. SCL low-hold release and ACK response on 9th SCL.
1	0	Command recovery mode	ACK response on 9th SCL and no SCL low-hold.
1	1	EEP response mode	NACK response on 9th SCL and no SCL low-hold.

#### WUF flag (Wakeup Event Occurrence Flag)

The WUF flag indicates whether the slave address is matching during wakeup.

[Setting condition]

- When PCLKB is supplied after the slave address of I<sup>2</sup>C bus is matched with the address whose enable bits of IC SER (except Device ID address) are set to 1 (enable), the bit is set.

[Clearing conditions]

- When 0 is written to the WUF flag after reading WUF = 1 (when WUSYF flag is set to 1).
- When ICCR1.ICE = 0 and IICRST = 1.

### 31.2.12 ICWUR2 : I<sup>2</sup>C Bus Wakeup Unit Register 2

Base address: IIC0WU = 0x4025\_E014  
IIC0WU\_NS = 0x5025\_E014

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	WUSY F	WUAS YF	WUSE N
Value after reset:	1	1	1	1	1	1	0	1

Bit	Symbol	Function	R/W
0	WUSEN	Wakeup Function Synchronous Enable 0: IIC asynchronous circuit enable 1: IIC synchronous circuit enable	R/W
1	WUASYF	Wakeup Function Asynchronous Operation Status Flag 0: IIC synchronous circuit enable condition 1: IIC asynchronous circuit enable condition	R
2	WUSYF	Wakeup Function Synchronous Operation Status Flag 0: IIC asynchronous circuit enable condition 1: IIC synchronous circuit enable condition	R
7:3	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE3, P-TYPE3

#### WUSEN bit (Wakeup Function Synchronous Enable)

The WUSEN bit is used in combination with the WUASYF flag (or WUSYF flag) to switch between the PCLKB synchronous and asynchronous operation, when the wakeup effective function is enabled (ICWUR.WUE = 1).

The PCLKB operation switches from synchronous to asynchronous operation:

When the ICCR2.BBSY flag is 0, if 0 is written to the WUSEN bit while the WUASYF flag is 0, the reception occurs independently of the operation of PCLKB (with PCLKB stopped) after it switches to the PCLKB asynchronous operation, on wakeup event detection.

The PCLKB operation switches from asynchronous to synchronous operation:

- When 1 is written to the WUSEN bit, with the WUASYF flag at 1, when a wakeup event is detected. After writing 1, the WUASYF flag immediately becomes 0.
- When the stop condition is detected with a wakeup event undetected.

[Setting condition]

- When 1 is written to the WUSEN bit.
- ICCR1.ICE = 0 and IICRST = 1 (IIC reset)
- ICWUR.WUE = 0

[Clearing conditions]

- When 0 is written to the WUSEN bit.

#### WUASYF flag (Wakeup Function Asynchronous Operation Status Flag)

The WUASYF flag can place the IIC in PCLKB asynchronous operation when the wakeup effective function is enabled (ICWUR.WUE = 1).

[Setting condition]

- When the ICCR2.BBSY flag is 0, and the WUSEN bit is set to 0 with the ICWUR.WUE bit set to 1.

[Clearing conditions]

- When 1 is written to the WUSEN bit after detecting the wake-up event with ICWUR.WUE bit set to 1.

- When a stop condition is detected with WUSEN bit set to 1 before detecting the wake-up event with WUASY flag set to 1 with ICWUR.WUE bit set to 1.
- When you write 1 in the WUSEN bit with the WUASYF flag detected 1 and the wake-up event in the state of ICWUR.WUE = 1.
- ICCR1.ICE = 0 and ICCRST = 1 (ICC reset)
- ICWUR.WUE = 0.

### WUSYF flag (Wakeup Function Synchronous Operation Status Flag)

It is shown that IIC is in the PCLKB synchronous operation at wake-up effective function (ICWUR.WUE = 1). This flag is a value in which the WUASYF flag is always reserved.

[Setting conditions]

- When 1 is written to the WUSEN bit after detecting the wake-up event with ICWUR.WUE bit set to 1 with WUSYF flag set to 0 with ICWUR.WUE bit set to 1.
- When a stop condition is detected with WUSEN bit set to 1 before detecting the wake-up event with WUSYF flag set to 0 with ICWUR.WUE bit set to 1.
- ICCR1.ICE = 0 and ICCRST = 1 (ICC reset)
- ICWUR.WUE = 0.

[Clearing condition]

- When the ICCR2.BBSY flag is 0 with the ICWUR.WUE bit set to 1 after writing 0 to the WUSEN bit.

### 31.2.13 SARLy : Slave Address Register Ly (y = 0 to 2)

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x0A + 0x02 × y

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SVA[6:0]							SVA0

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	SVA0	10-bit Address LSB Slave address setting.	R/W
7:1	SVA[6:0]	7-bit Address/10-bit Address Lower Bits Slave address setting.	R/W

Note: S-TYPE3, P-TYPE3

#### SVA0 bit (10-bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 1), the SVA0 bit functions as the LSB of a 10-bit address and is combined with the SVA[6:0] bits to form the lower 8 bits of a 10-bit address.

This bit is valid when the SARyE bit in ICSEr is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in this bit is ignored.

#### SVA[6:0] bits (7-bit Address/10-bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS = 0), the SVA[6:0] bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits combine with the SVA0 bit to form the lower 8 bits of a 10-bit address.

When the SARyE bit in ICSEr is 0, the setting in these bits is ignored.

### 31.2.14 SARUy : Slave Address Register Uy (y = 0 to 2)

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x0B + 0x02 × y

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SVA[1:0]		FS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FS	7-bit/10-bit Address Format Select 0: Select 7-bit address format 1: Select 10-bit address format	R/W
2:1	SVA[1:0]	10-bit Address Upper Bits Slave address setting.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

#### FS bit (7-bit/10-bit Address Format Select)

The FS bit selects 7- or 10-bit format for slave address y (in SARLy and SARUy).

When the SARyE bit in ICSEr is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the SVA[1:0] and SVA0 settings in SARLy are ignored.

When the SARyE bit in ICSEr is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the SVA[1:0] and SARLy settings are valid.

When the SARyE bit in ICSEr is 0 (SARLy and SARUy disabled), the SARUy.FS setting is invalid.

#### SVA[1:0] bits (10-bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), the SVA[1:0] bits function as the upper 2 bits of a 10-bit address.

These bits are valid when the SARyE bit in ICSEr is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in these bits is ignored.

### 31.2.15 ICBRL : I<sup>2</sup>C Bus Bit Rate Low-Level Register

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	BRL[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	BRL[4:0]	Bit Rate Low-Level Period Low-level period of SCL clock.	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE3, P-TYPE3

#### BRL[4:0] bits (Bit Rate Low-Level Period)

The BRL[4:0] bits set the low-level period of the SCL clock. ICBRL counts the low-level period with the internal reference clock source (IICφ) specified by the CKS[2:0] bits in ICMR1. ICBRL also generates the data setup time for automatic SCL

low-hold operation, see [section 31.9. Automatic Low-Hold Function for SCL](#). When the IIC is used in slave mode, the BRL[4:0] bits must be set to a value longer than the data setup time\*<sup>1</sup>.

If the digital noise filter is enabled (NFE bit in ICFER is 1), set the BRL[4:0] bits to a value at least one greater than the number of stages in the noise filter. For details on the number of stages, see the description of the NF[1:0] bits in [section 31.2.5. ICMR3 : I<sup>2</sup>C Bus Mode Register 3](#).

Note 1. Data setup time (t<sub>SU</sub>: DAT)

250 ns for up to 100 kbps: Standard-mode (Sm)

100 ns for up to 400 kbps: Fast-mode (Fm)

50 ns for up to 1 Mbps: Fast-mode Plus (Fm+)

### 31.2.16 ICBRH : I<sup>2</sup>C Bus Bit Rate High-Level Register

Base address: IICn = 0x4025\_E000 + 0x0100 × n (n = 0 to 1)  
IICn\_NS = 0x5025\_E000 + 0x0100 × n (n = 0 to 1)

Offset address: 0x11

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	BRH[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	BRH[4:0]	Bit Rate High-Level Period High-level period of SCL clock.	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE3, P-TYPE3

#### BRH[4:0] bits (Bit Rate High-Level Period)

The BRH[4:0] bits set the high-level period of SCL clock. BRH[4:0] bits are valid in master mode. If the IIC is used only in slave mode, do not set the BRH[4:0] bits.

ICBRH counts the high-level period with the internal reference clock source (IICφ) specified in the CKS[2:0] bits in ICMR1.

If the digital noise filter is enabled (the NFE bit in ICFER is 1), set these bits to a value at least one greater than the number of stages in the noise filter. For the number of stages in the noise filter, see the description of the NF[1:0] bits in [section 31.2.5. ICMR3 : I<sup>2</sup>C Bus Mode Register 3](#).

The IIC transfer rate and the SCL clock duty are calculated using the following expressions (1) to (5):

1. ICFER.SCLE = 0

$$\text{Transfer rate} = 1 / [ \{ (BRH + 1) + (BRL + 1) \} / IIC\phi^{*1} + tr^{*2} + tf^{*2} ]$$

$$\text{Duty cycle} = [ tr + \{ (BRH + 1) / IIC\phi \} ] / [ tr + tf + \{ (BRH + 1) + (BRL + 1) \} / IIC\phi ]$$

2. ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] = 000b (IICφ = PCLKB)

$$\text{Transfer rate} = 1 / [ \{ (BRH + 3) + (BRL + 3) \} / IIC\phi + tr + tf ]$$

$$\text{Duty cycle} = [ tr + \{ (BRH + 3) / IIC\phi \} ] / [ tr + tf + \{ (BRH + 3) + (BRL + 3) \} / IIC\phi ]$$

3. ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] = 000b (IICφ = PCLKB)

$$\text{Transfer rate} = 1 / [ \{ (BRH + 3 + nf^{*3}) + (BRL + 3 + nf) \} / IIC\phi + tr + tf ]$$

$$\text{Duty cycle} = [ tr + \{ (BRH + 3 + nf) / IIC\phi \} ] / [ tr + tf + \{ (BRH + 3 + nf) + (BRL + 3 + nf) \} / IIC\phi ]$$

4. ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] ≠ 000b

$$\text{Transfer rate} = 1 / [ \{ (BRH + 2) + (BRL + 2) \} / IIC\phi + tr + tf ]$$

$$\text{Duty cycle} = [ tr + \{ (BRH + 2) / IIC\phi \} ] / [ tr + tf + \{ (BRH + 2) + (BRL + 2) \} / IIC\phi ]$$

5. ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] ≠ 000b

$$\text{Transfer rate} = 1 / [ \{ (BRH + 2 + nf) + (BRL + 2 + nf) \} / IIC\phi + tr + tf ]$$

$$\text{Duty cycle} = [ tr + \{ (BRH + 2 + nf) / IIC\phi \} ] / [ tr + tf + \{ (BRH + 2 + nf) + (BRL + 2 + nf) \} / IIC\phi ]$$



Note 1.  $IIC\phi = PCLKB \times \text{division ratio}$

Note 2. The SCLn line rise time (tr) and SCLn line fall time (tf) depend on the total bus line capacitance (Cb) and the pull-up resistor (Rp). For details, see the I2C bus standard from NXP Semiconductors.

Note 3. nf = Number of digital noise filters selected in the ICMR3.NF bit.

**Table 31.6 Example of ICBRH/ICBRL Settings for Transfer Rate IIC when SCLE = 0**

Transfer rate (kbps)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	100b	14 (0xEE)	17 (0xF1)	60	—	(1)
400	010b	8 (0xE8)	19 (0xF3)	60	—	(1)
1000	000b	15 (0xEF)	29 (0xFD)	60	—	(1)

**Table 31.7 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 0**

Transfer rate (kbps)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	100b	13 (0xED)	16 (0xF0)	60	—	(4)
400	010b	7 (0xE7)	18 (0xF2)	60	—	(4)
1000	000b	13 (0xED)	27 (0xFB)	60	—	(2)

**Table 31.8 Example of ICBRH/ICBRL Settings for Transfer Rate when SCLE = 1 and NFE = 1**

Transfer rate (kbps)	CKS[2:0] (ICMR1)	BRH[4:0] (ICBRH)	BRL[4:0] (ICBRL)	PCLKB (MHz)	NF[1:0]	Computation expression
100	100b	11 (0xEB)	14 (0xEE)	60	01b	(5)
400	010b	5 (0xE5)	16 (0xF0)	60	01b	(5)
1000	000b	11 (0xEB)	25 (0xF9)	60	01b	(3)

### 31.2.17 ICDRT : I2C Bus Transmit Data Register

Base address:  $IICn = 0x4025\_E000 + 0x0100 \times n$  (n = 0 to 1)  
 $IICn\_NS = 0x5025\_E000 + 0x0100 \times n$  (n = 0 to 1)

Offset address: 0x12

Bit position: 7 6 5 4 3 2 1 0

Bit field:

Value after reset: 1 1 1 1 1 1 1 1

When ICDRT detects a space in the I2C Bus Shift Register (ICDRS), it transfers the transmit data that was written to ICDRT to ICDRS and starts transmitting data in transmit mode. The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data is written to ICDRT while the ICDRS data is being transmitted.

ICDRT can always be read and written to. Write transmit data to ICDRT once when a transmit data empty interrupt (IICn\_TXI) request is generated.

### 31.2.18 ICDRR : I2C Bus Receive Data Register

Base address:  $IICn = 0x4025\_E000 + 0x0100 \times n$  (n = 0 to 1)  
 $IICn\_NS = 0x5025\_E000 + 0x0100 \times n$  (n = 0 to 1)

Offset address: 0x13

Bit position: 7 6 5 4 3 2 1 0

Bit field:

Value after reset: 0 0 0 0 0 0 0 0

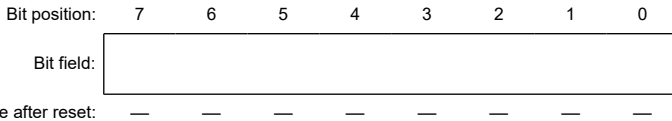
When 1 byte of data is received, the received data is transferred from the I<sup>2</sup>C Bus Shift Register (ICDRS) to ICDRR to enable the next data to be received. The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data is read from ICDRR while ICDRS is receiving data. ICDRR cannot be written to. Read data from ICDRR once when a receive data full interrupt (IICn\_RXI) request is generated.

If ICDRR receives the next receive data before the current data is read from ICDRR (while the RDRF flag in ICSR2 is 1), the IIC automatically holds the SCL clock low 1 cycle before the RDRF flag is set to 1 next.

### 31.2.19 ICDRS : I<sup>2</sup>C Bus Shift Register

Base address: n/a

Offset address: n/a



ICDRS is an 8-bit shift register for data transmit and receive. During transmission, transmit data is transferred from ICDRT to ICDRS and is transmitted from the SDAn pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data is received. ICDRS cannot be accessed directly.

## 31.3 Operation

### 31.3.1 Communication Data Format

The I<sup>2</sup>C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start or restart condition is an address frame that specifies a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.

Figure 31.3 shows the I<sup>2</sup>C bus format, and Figure 31.4 shows the I<sup>2</sup>C bus timing.

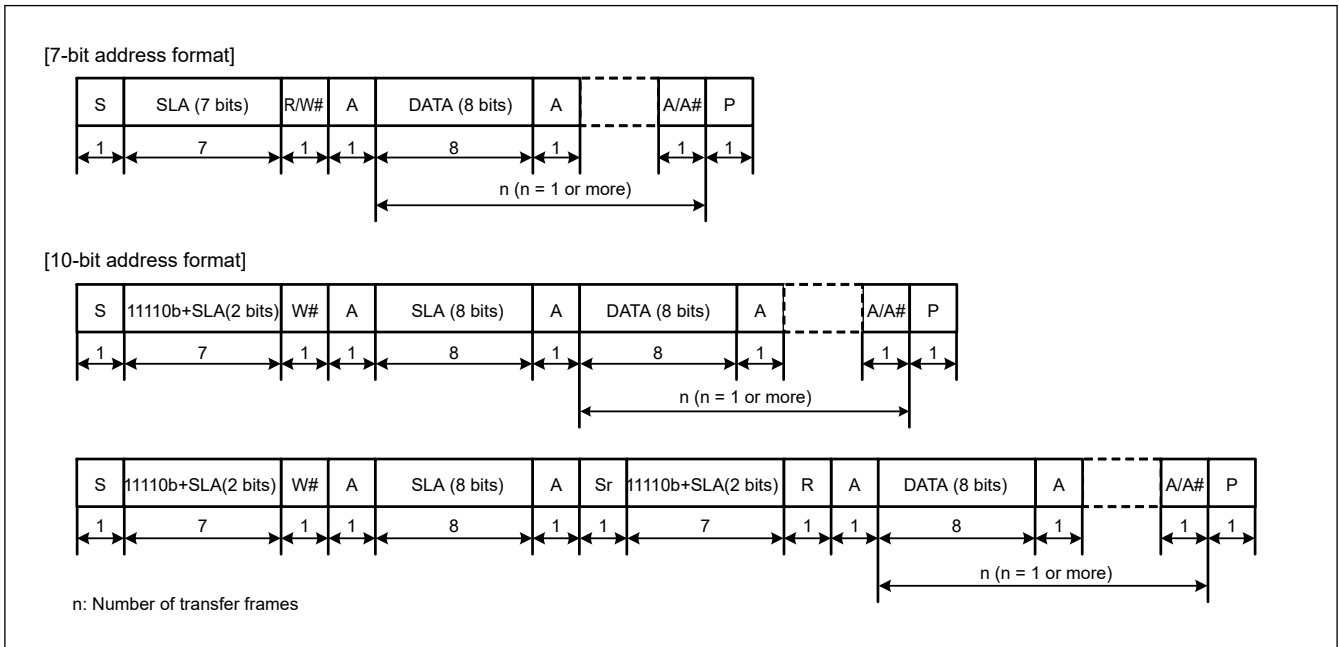
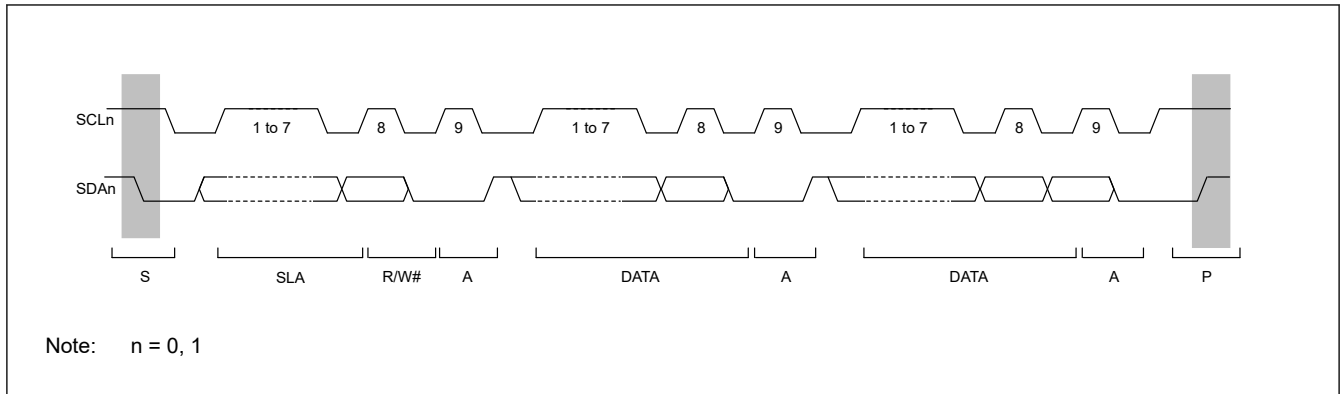


Figure 31.3 I<sup>2</sup>C bus format



**Figure 31.4** I<sup>2</sup>C bus timing when the SLA setting = 7 bits

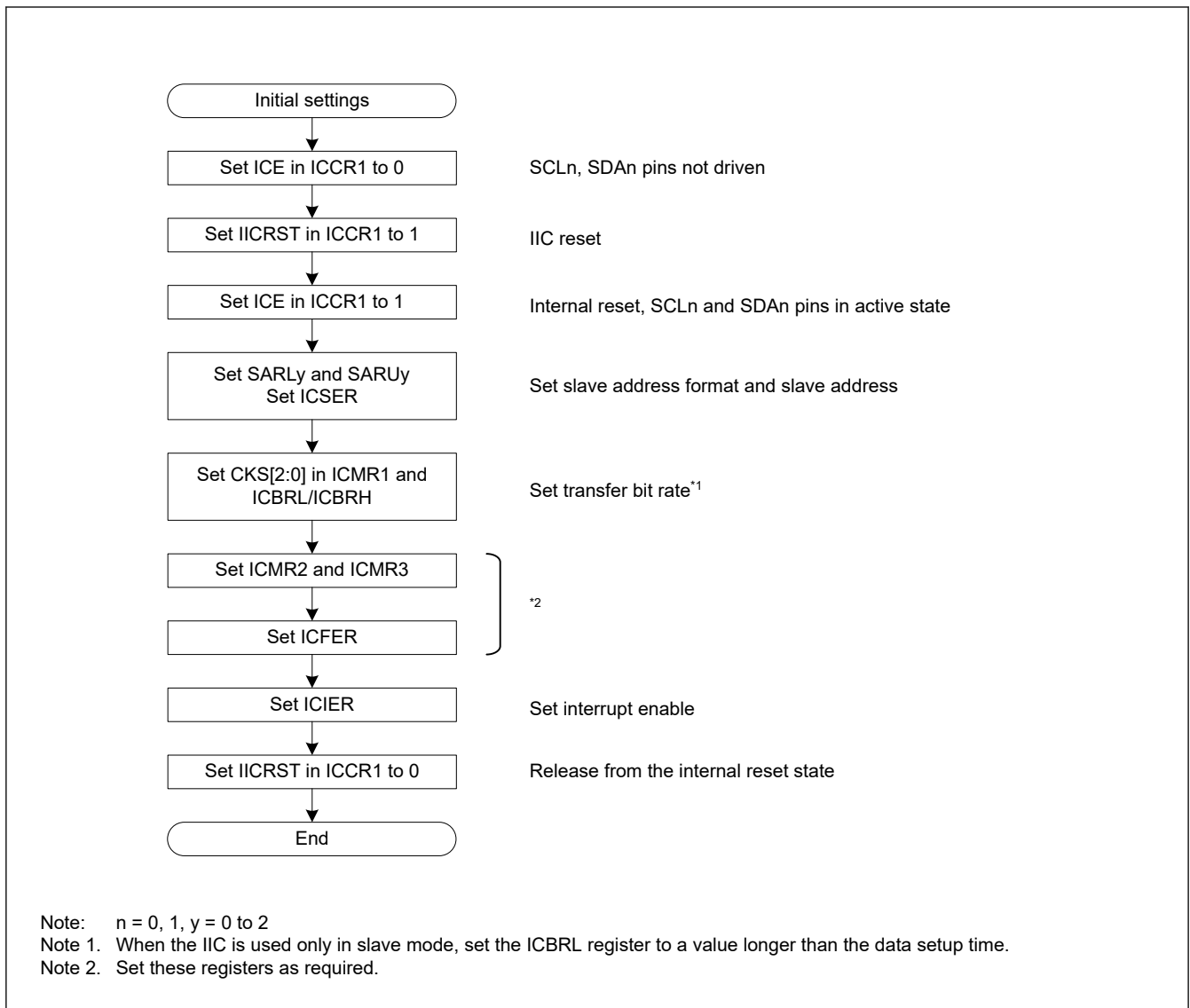
- S: Start condition. The master device drives the SDAn line low from high while the SCLn line is high.
- SLA: Slave address, by which the master device selects a slave device.
- R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W# is 1, or from the master device to the slave device when R/W# is 0.
- A: Acknowledge. The receive device drives the SDAn line low. In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.
- A#: Not Acknowledge. The receive device drives the SDAn line high.
- Sr: Restart condition. The master device drives the SDAn line low from the high level after the setup time has elapsed with the SCLn line high.
- DATA: Transmitted or received data.
- P: Stop condition. The master device drives the SDAn line high from low while the SCLn line is high.

### 31.3.2 Initial Settings

Before starting data transmission or reception, initialize the IIC using the procedure shown in [Figure 31.5](#).

1. Set the ICCR1.ICE bit to 0 to set the SCLn and SDAn pins to the inactive state.
2. Set the ICCR1.IICRST bit to 1 to initiate IIC reset.
3. Set the ICCR1.ICE bit to 1 to initiate internal reset.
4. Set the SARLy, SARUy, ICSEr, ICMR1, ICBRH, and ICBRL registers ( $y = 0$  to 2), and set the other registers as required. For initial settings of the IIC, see [Figure 31.5](#).
5. When the required register settings are complete, set the ICCR1.IICRST bit to 0 to release the IIC reset.

This procedure is not required if the IIC initialization is already complete.



**Figure 31.5 Example IIC initialization flow**

### 31.3.3 Master Transmit Operation

In master transmit operation, the IIC outputs the SCL clock and transmitted data signals as the master device, and the slave device returns acknowledgments. [Figure 31.6](#) shows an example of master transmission, and [Figure 31.7](#) to [Figure 31.9](#) show the operation timing in master transmission.

To set up and perform master transmission:

1. Process initial settings. For details, see [section 31.3.2. Initial Settings](#).
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and START flags in ICSR2 automatically set to 1 and the ST bit automatically is set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically is set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag automatically is set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. After the byte containing the slave address and R/W# bit is transmitted, the value of the TRS bit automatically updates to select master transmit or master receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the IIC continues in master transmit mode.

Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.

For data transmission with an address in the 10-bit format, start by writing 11110b, the 2 higher-order bits of the slave address, and W to ICDRT as the first address transmission. For the second address transmission, write the 8 lower-order bits of the slave address to ICDRT.

4. Check that the TDRE flag in ICSR2 is 1, and then write the transmit data to the ICDRT register. The IIC automatically holds the SCLn line low until the transmit data is ready or a stop condition is issued.
5. After all bytes of transmit data are written to the ICDRT register, wait until the value in the TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 to 1 (stop condition requested). On receiving a stop condition request, the IIC issues the stop condition. Regarding issuing a stop condition, see [section 31.11.3. Issuing a Stop Condition](#).
6. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, it automatically sets the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
7. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

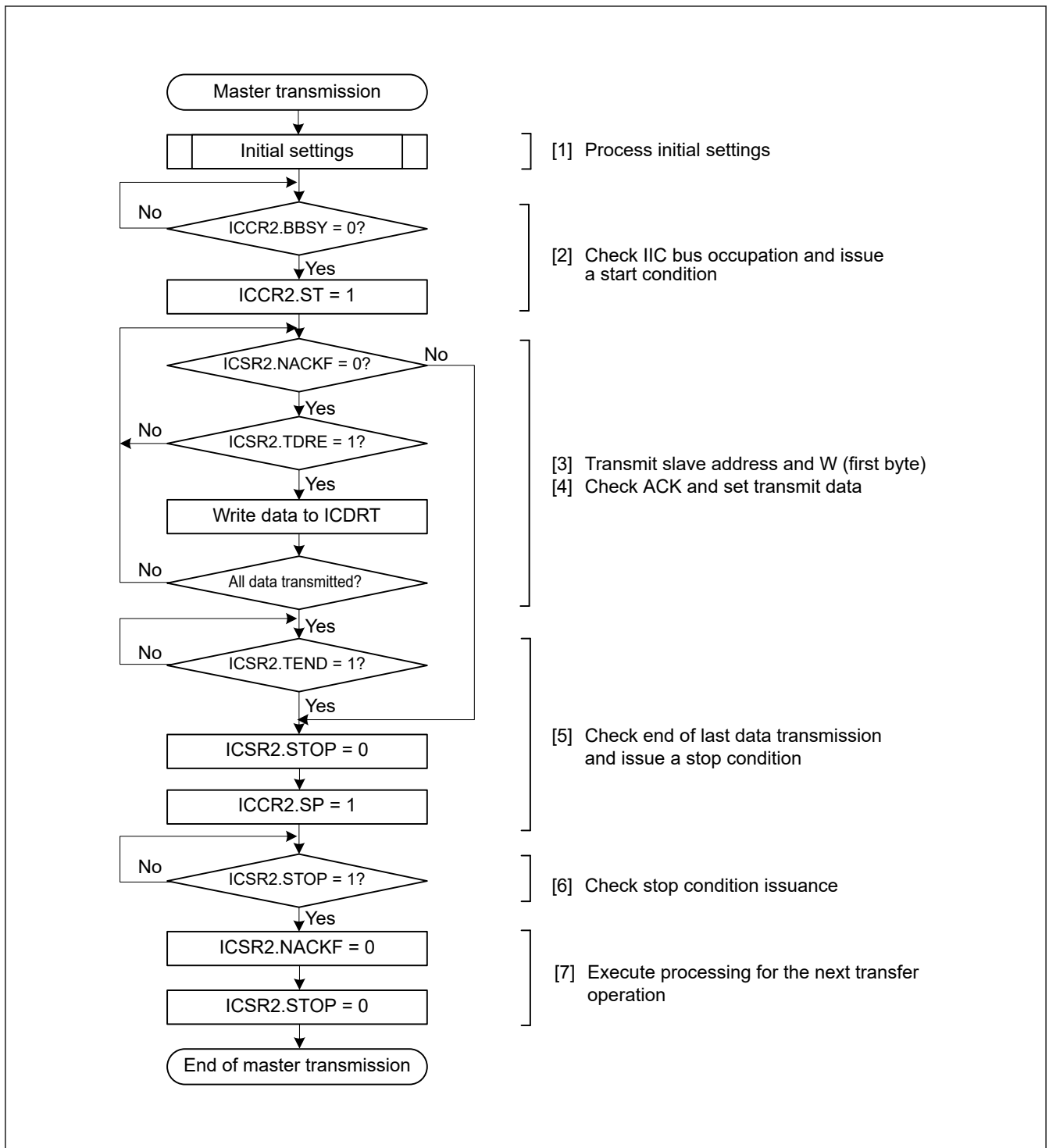


Figure 31.6 Example master transmission flow

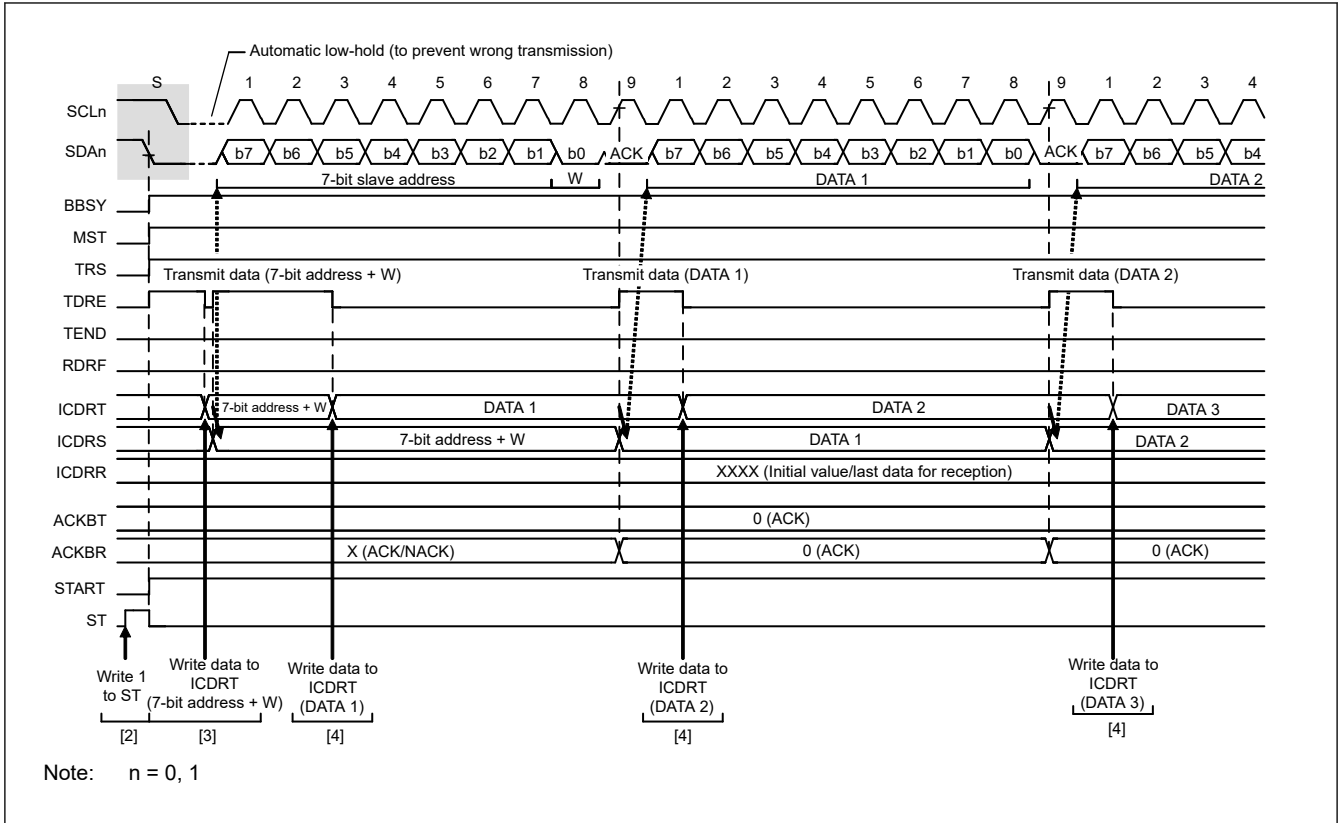


Figure 31.7 Master transmit operation timing (1) with 7-bit address format

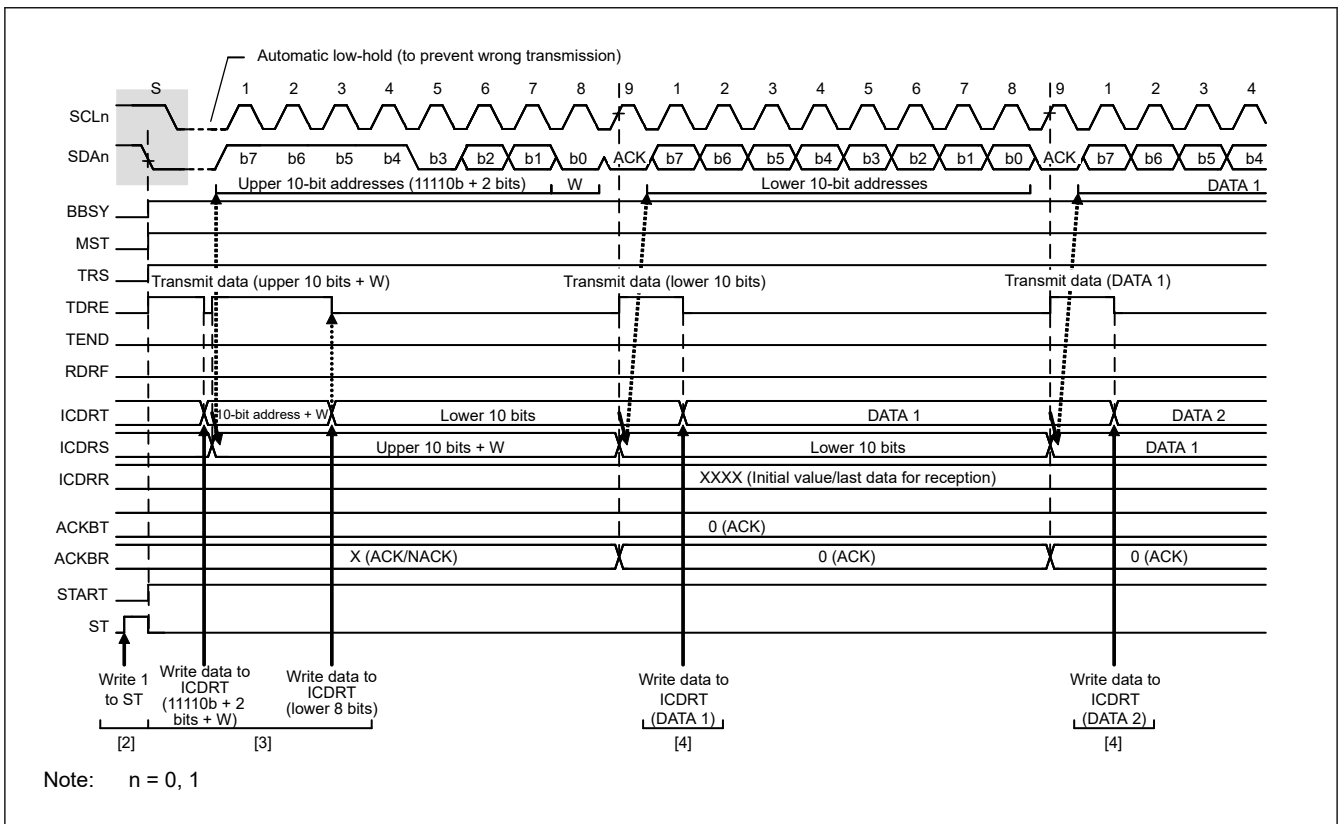


Figure 31.8 Master transmit operation timing (2) with 10-bit address format

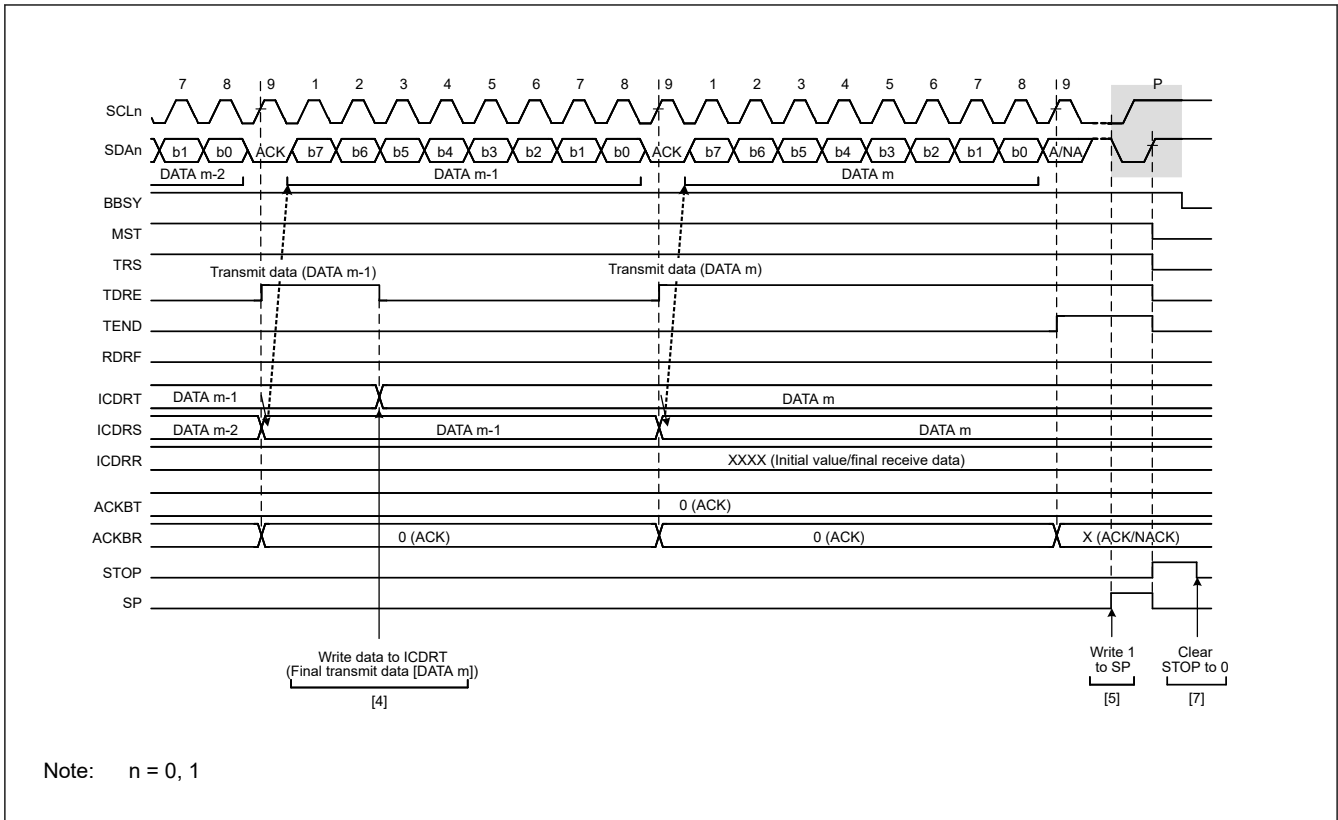


Figure 31.9 Master transmit operation timing (3)

### 31.3.4 Master Receive Operation

In master receive operation, the IIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because the IIC must start by sending a slave address to the associated slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 31.10 and Figure 31.11 show examples of master reception (7-bit address format), and Figure 31.12 to Figure 31.14 show the operation timing in master reception.

To set up and perform master reception:

1. Process initial settings. For details, see section 31.3.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. When the IIC detects the start condition, the BBSY and START flags in ICSR2 automatically set to 1, and the ST bit automatically is set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA<sub>n</sub> line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically is set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag automatically is set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. When the byte containing the slave address and R/W# bit is transmitted, the value of the ICCR2.TRS bit automatically updates to select transmit or receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit is 1, the TRS bit is set to 0 on the rising edge of the ninth cycle of the SCL clock, placing the IIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag automatically is set to 1. Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition. For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmitting 11110b, the two higher-order bits of the slave address, and the R bit places the IIC in master receive mode.



4. Dummy read ICDRR after confirming that the RDRF flag in ICSR2 is 1. This makes the IIC start output of the SCL clock and start data reception.
5. After 1 byte of data is received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the 8th or 9th cycle of the SCL clock, as selected in the RDRFS bit in ICMR3. Reading ICDRR at this time produces the received data, and the RDRF flag is automatically set to 0 at the same time. Additionally, the value of the acknowledgment field received during the ninth cycle of the SCL clock is returned as the value set in the ICMR3.ACKBT bit. If the next byte to be received is the second-to-last byte, set the ICMR3.WAIT bit to 1 for wait insertion before reading ICDRR, containing the second-to-last byte. In addition to enabling NACK output, even when interrupts or other operations result in delays in setting the ICMR3.ACKBT bit to 1 (NACK) in step (6), this fixes the SCLn line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, which enables the issuing of a stop condition.
6. When the ICMR3.RDRFS bit is 0, and the slave device must be notified that it is to end transfer for data reception after transfer of the next and final byte, set the ICMR3.ACKBT bit to 1 (NACK).
7. After reading the second-to-last byte from the ICDRR register, if the value of the ICSR2.RDRF flag is 1, write 1 to the SP bit in ICCR2 (stop condition requested), and then read the last byte from ICDRR. When ICDRR is read, the IIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is complete or the SCLn line is released from the low-hold state.
8. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, detection of the stop condition sets the ICSR2.STOP flag to 1.
9. Check that the ICSR2.STOP flag is 1, then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

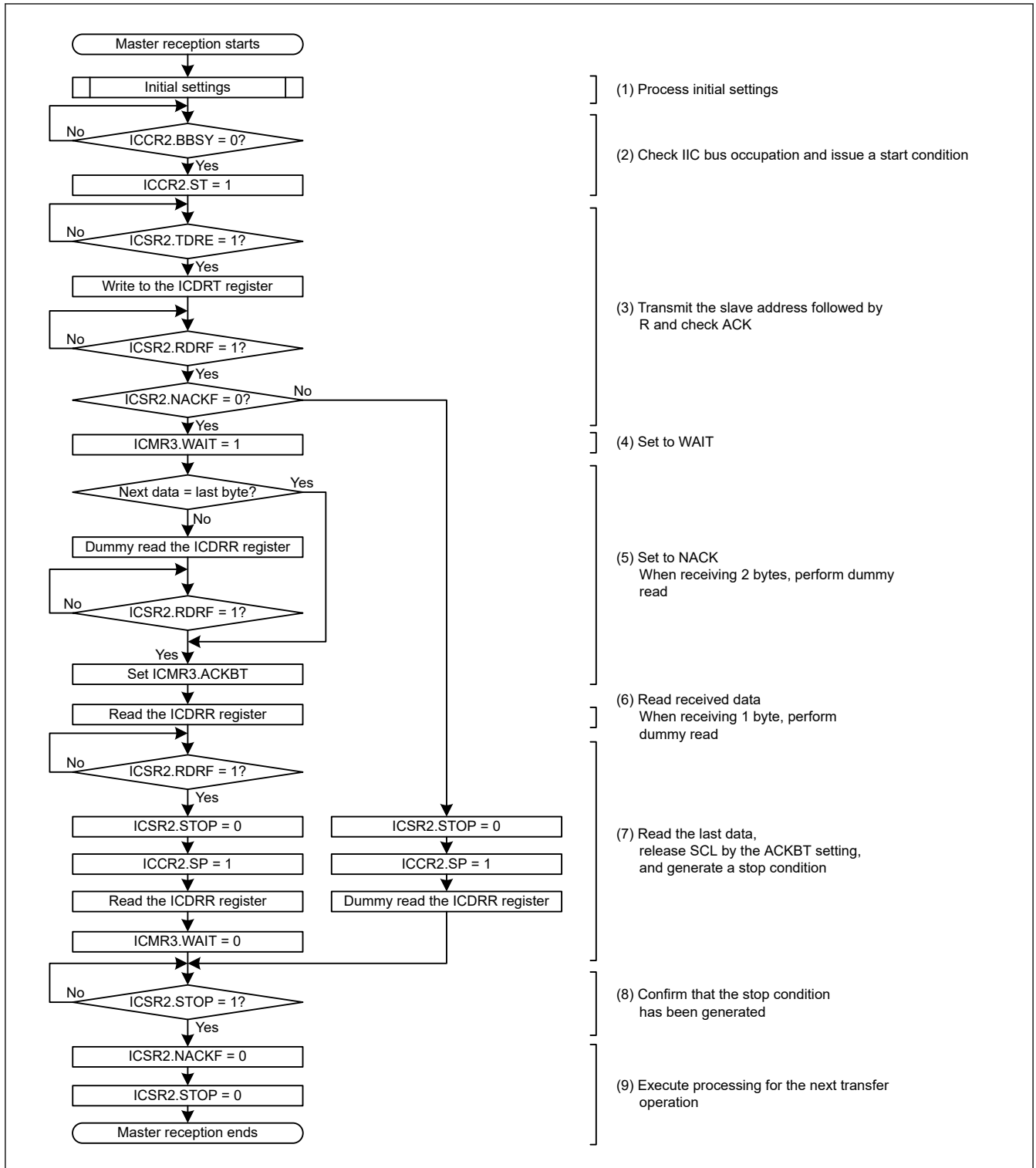


Figure 31.10 Example master reception flow with 7-bit address format of 1 byte or 2 bytes

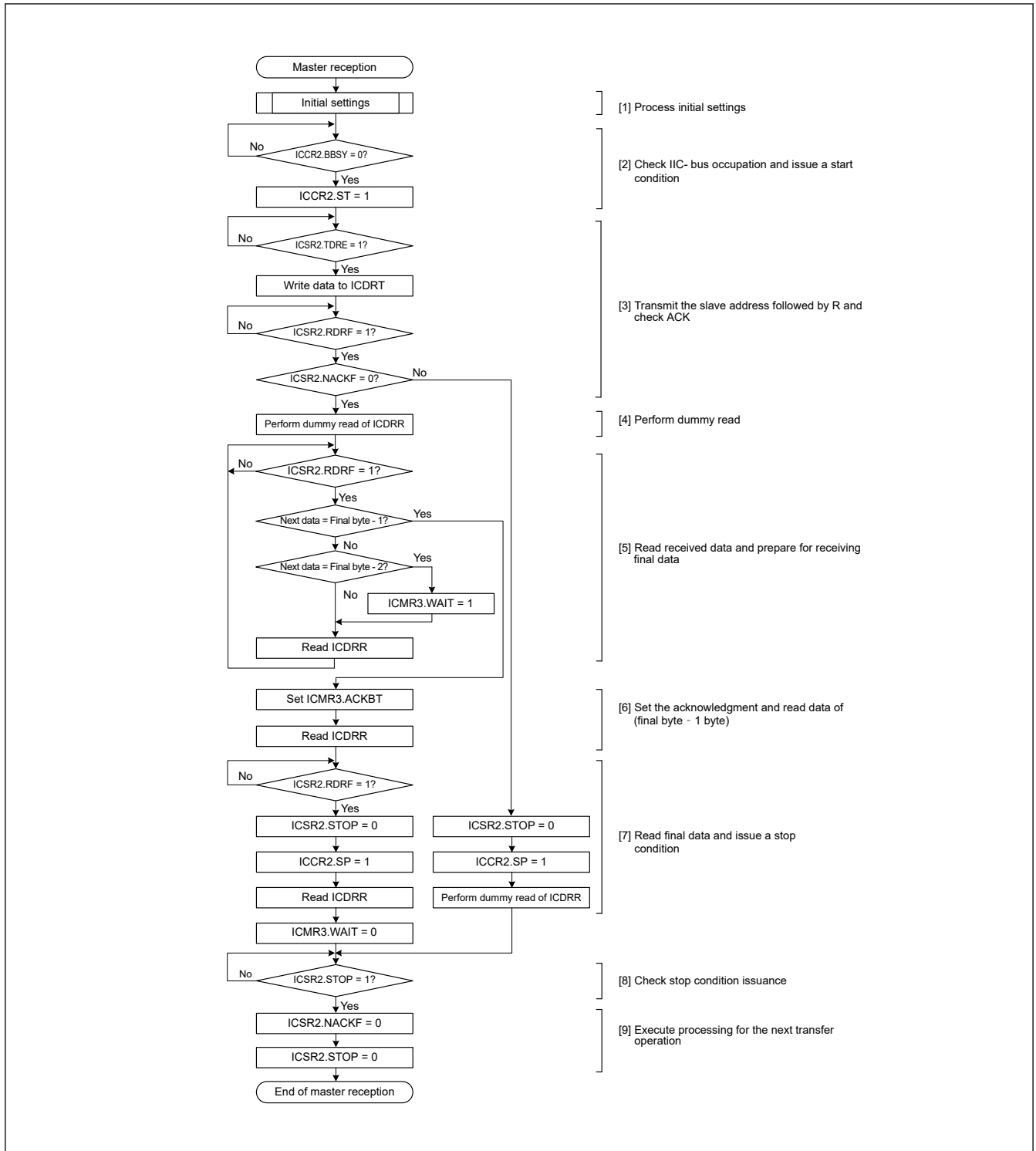


Figure 31.11 Example master reception flow with 7-bit address format of 3 or more bytes

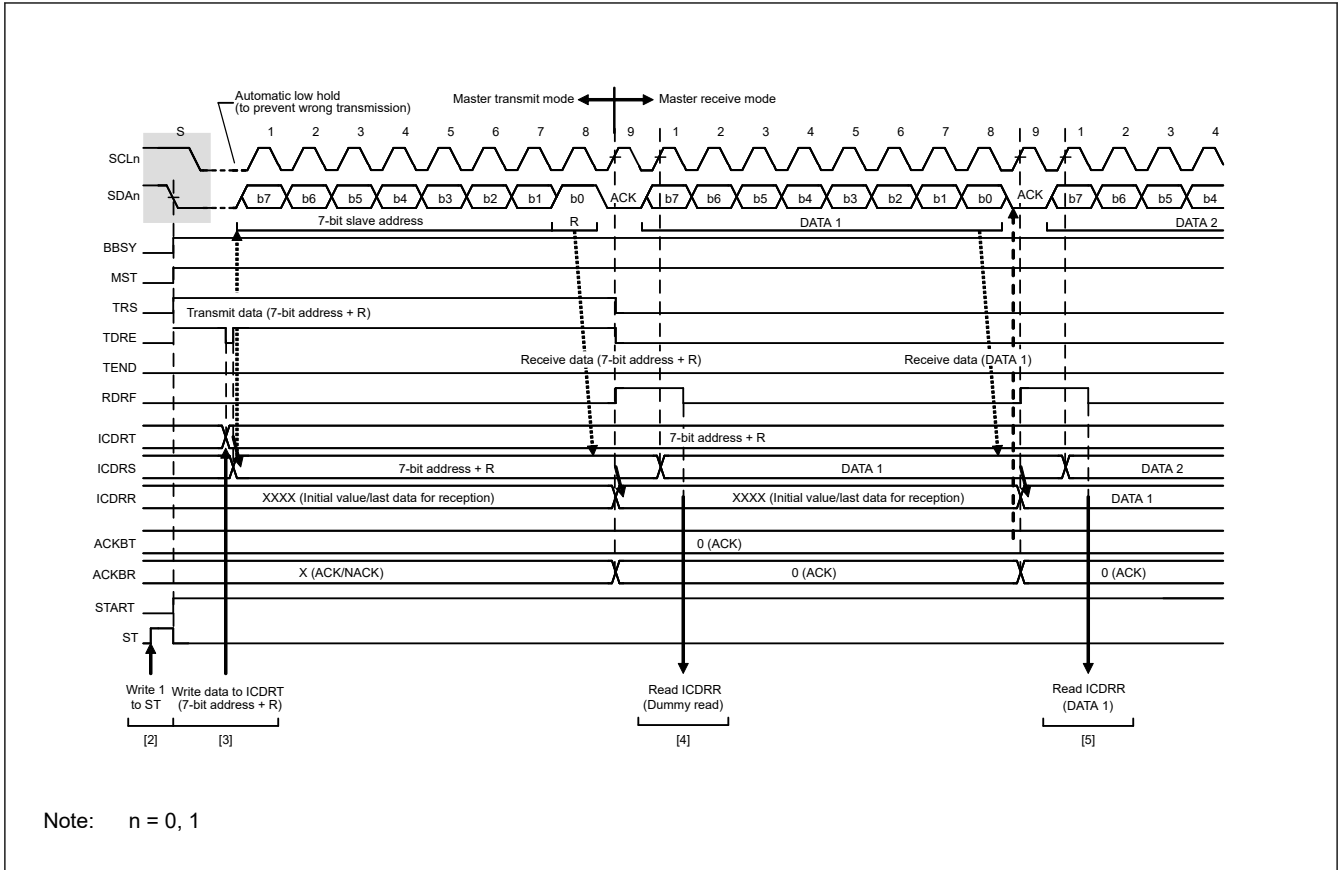


Figure 31.12 Master receive operation timing (1) with 7-bit address format when RDRFS = 0

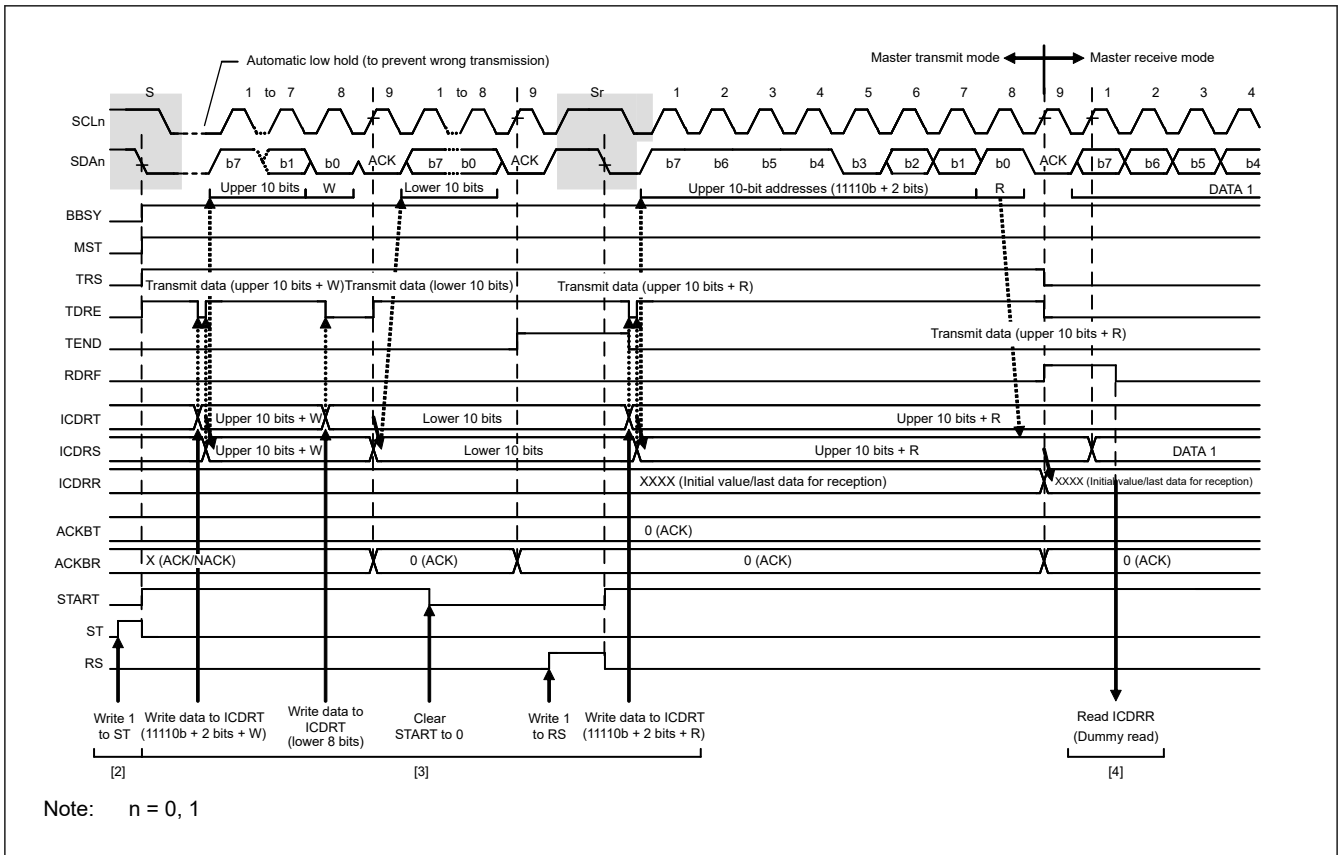


Figure 31.13 Master receive operation timing (2) with 10-bit address format when RDRFS = 0

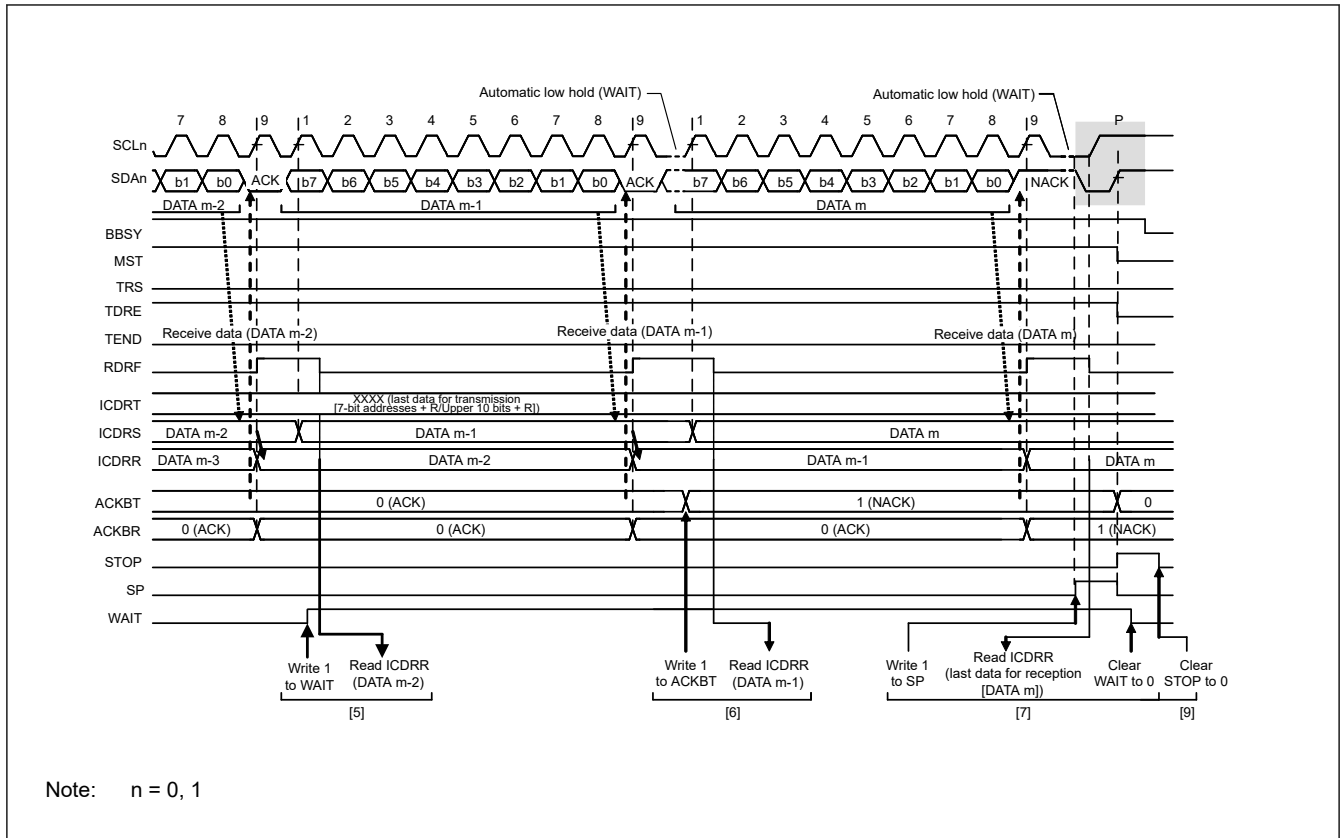


Figure 31.14 Master receive operation timing (3) when RDRFS = 0

### 31.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL clock, the IIC transmits data as a slave device, and the master device returns acknowledgments.

Figure 31.15 shows an example of slave transmission, and Figure 31.16 and Figure 31.17 show the operation timing in slave transmission.

To set up and perform slave transmission:

1. Initialize the IIC using the procedure in section 31.3.2. Initial Settings.  
After initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 1 on the rising edge of the ninth cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of the SCL clock. If the value of the R/W# bit is 1, the IIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.
3. Check that the ICSR2.TDRE flag is 1, then write the transmit data to the ICDRT register. If the IIC receives no acknowledge from the master device (receives an NACK signal) while the ICFER.NACKF bit is 1, the IIC suspends transfer of the next data.
4. Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the IIC drives the SCLn line low on the ninth falling edge of the SCL clock.
5. When the ICSR2.NACKF or ICSR2.TEND flag is 1, dummy read ICDRR to complete the processing. This releases the SCLn line.
6. On detecting the stop condition, the IIC automatically sets the ICSR1.HOA, GCA, and AASy flags (y = 0 to 2), the ICSR2.TDRE and TEND flags, and the ICCR2.TRS bit to 0, and enters slave receive mode.
7. Check that the ICSR2.STOP flag is 1, then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

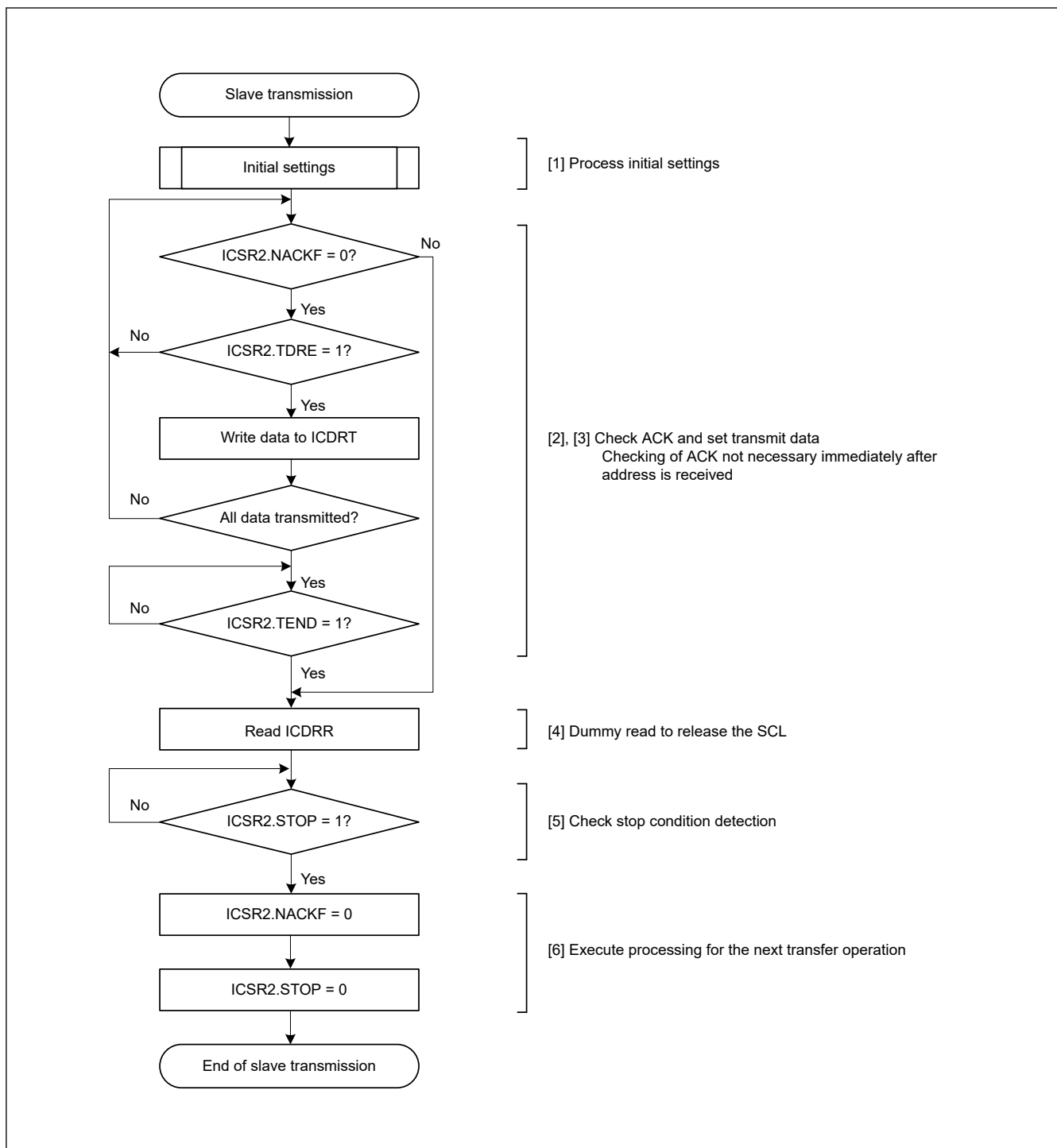


Figure 31.15 Example slave transmission flow

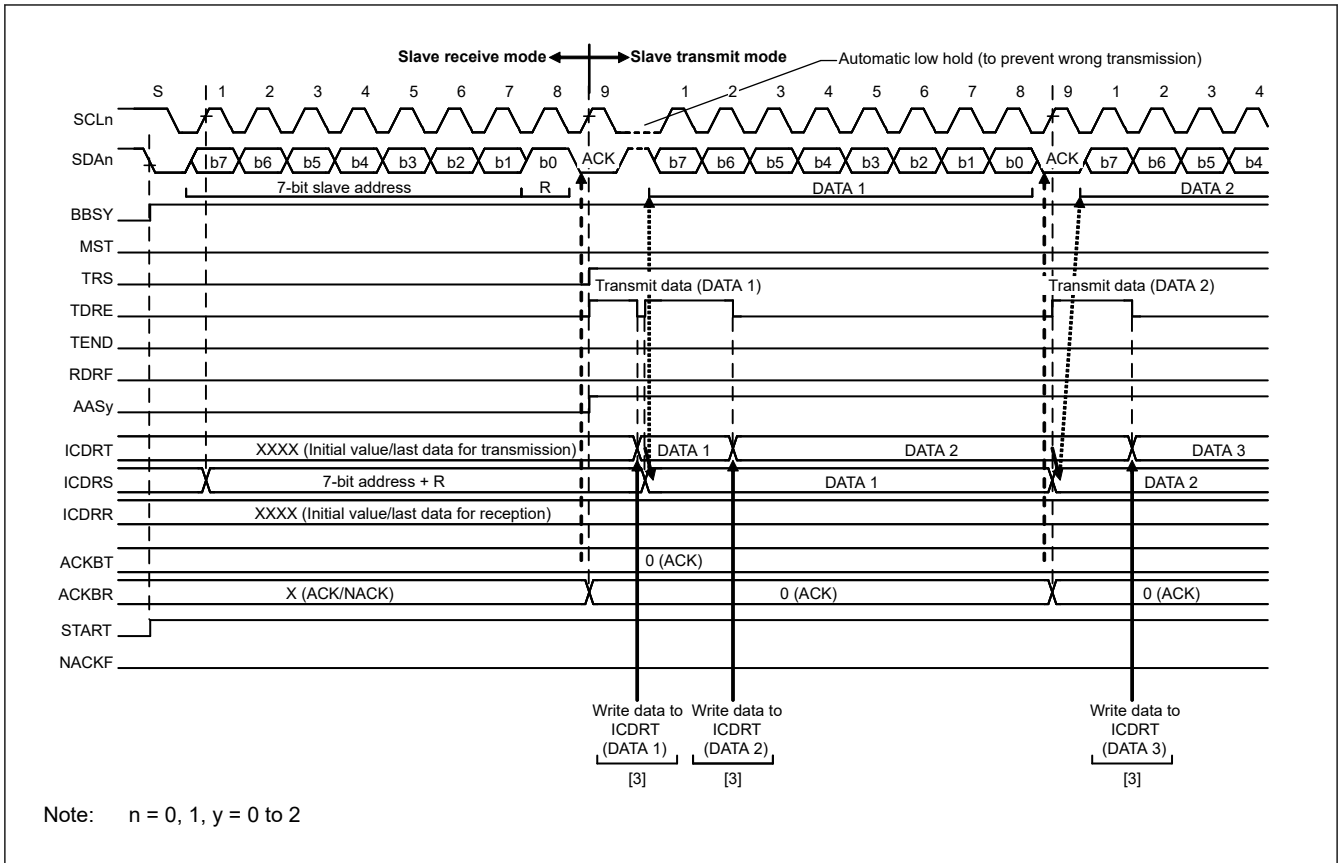


Figure 31.16 Slave transmit operation timing (1) with 7-bit address format

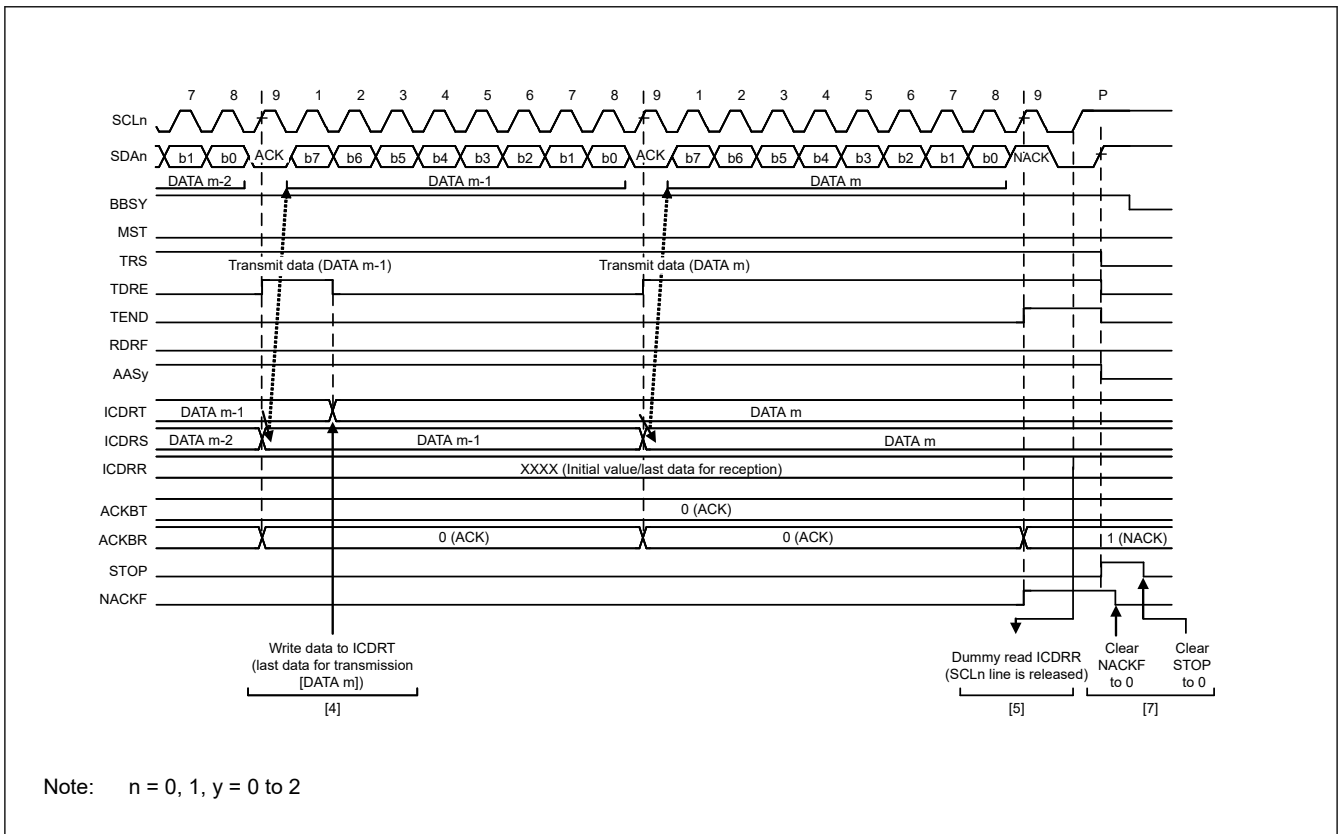


Figure 31.17 Slave transmit operation timing (2)

### 31.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the IIC returns acknowledgments as a slave device.

Figure 31.18 shows an example of slave reception, and Figure 31.19 and Figure 31.20 show the operation timing in slave reception.

To set up and perform slave reception:

1. Initialize the IIC using the procedure in [section 31.3.2. Initial Settings](#).  
After initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 1 on the rising edge of the ninth cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of the SCL clock. If the value of the R/W# bit is 0, the IIC continues to place itself in slave receive mode and sets the RDRF flag in ICSR2 to 1.
3. Check that the ICSR2.STOP flag is 0 and the ICSR2.RDRF flag is 1, then dummy read ICDRR. The dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected.
4. When ICDRR is read, the IIC automatically sets the ICSR2.RDRF flag to 0. If reading of ICDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the IIC holds the SCLn line low until 1 SCL cycle before the point where RDRF must be set. In this case, reading ICDRR releases the SCLn line from being held low. When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read ICDRR until all the data is completely received.
5. On detecting the stop condition, the IIC automatically clears the ICSR1.HOA, GCA, and AASy flags (y = 0 to 2) to 0.
6. Check that the ICSR2.STOP flag is 1, then set the ICSR2.STOP flag to 0 for the next transfer operation.

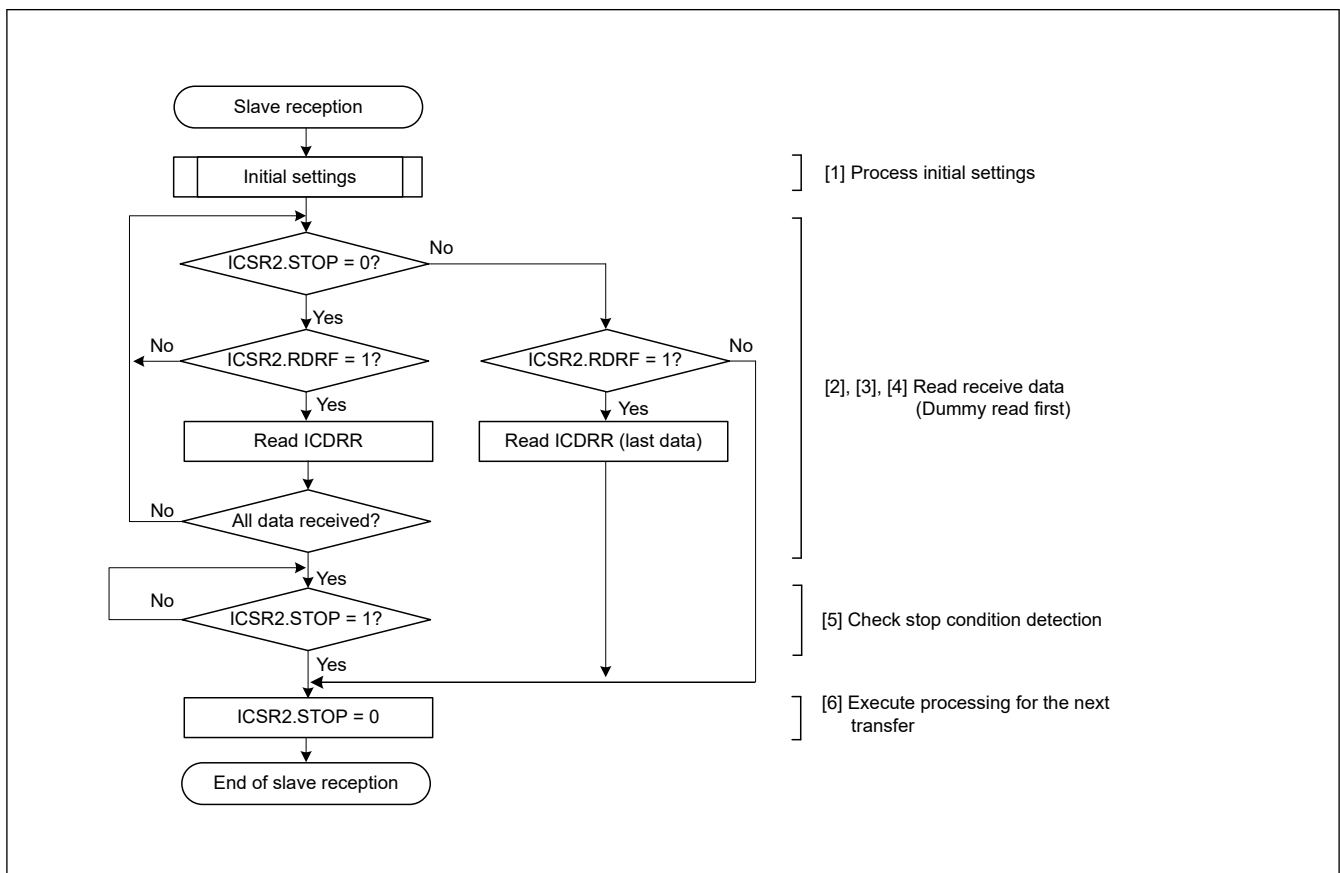


Figure 31.18 Example slave reception flow



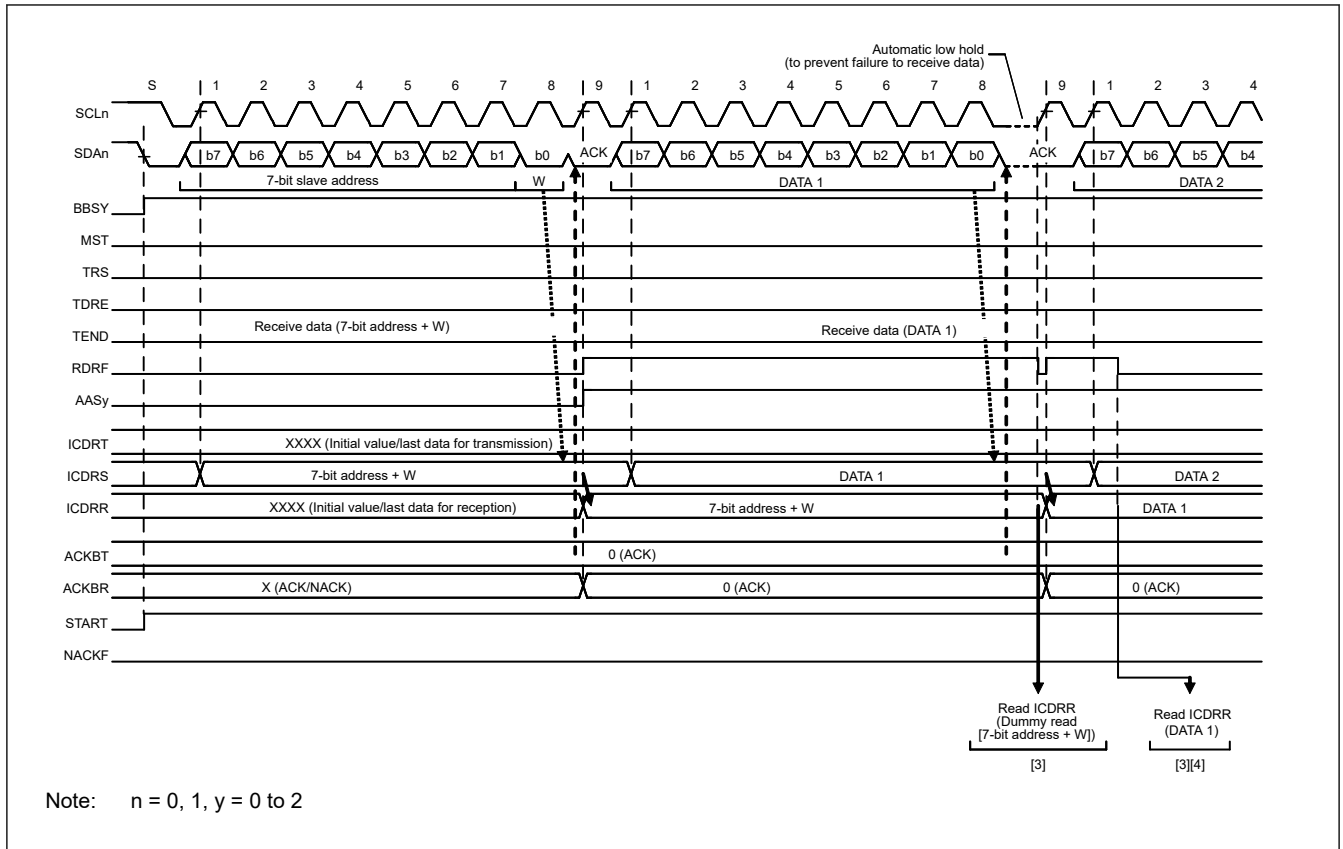


Figure 31.19 Slave receive operation timing (1) with 7-bit address format when RDRFS = 0

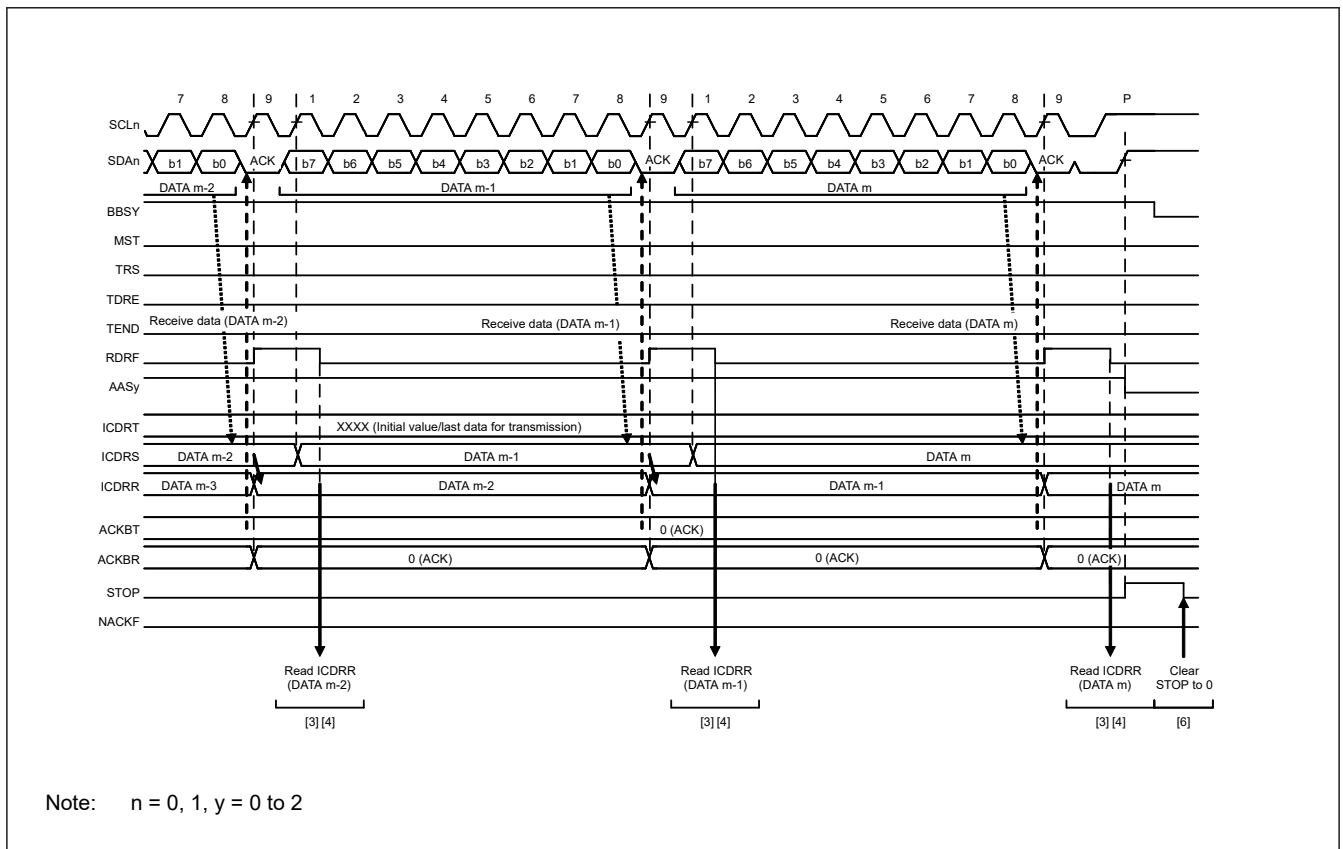


Figure 31.20 Slave receive operation timing (2) when RDRFS = 0

### 31.4 SCL Synchronization Circuit

For generation of the SCL clock, the IIC starts counting the value for the high-level period specified in ICBRH when it detects a rising edge on the SCLn line, and it drives the SCLn line low when it completes counting. When the IIC detects the falling edge of the SCLn line, it starts counting the value for the low-level period specified in ICBRL, and then it stops driving the SCLn line, releasing the line, when it completes counting. The IIC repeats this process to generate the SCL clock.

If multiple master devices are connected to the I<sup>2</sup>C bus, a collision of SCL signals might arise because of contention with another master device. In such cases, the master devices must synchronize their SCL signals. Because this synchronization of SCL signals must be bit-by-bit, the IIC is equipped with an SCL synchronization circuit to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When the IIC detects a rising edge on the SCLn line and starts counting the high-level period specified in ICBRH.BRH[4:0], and the level on the SCLn line falls because an SCL signal is being generated by another master device, the IIC performs the following:

1. Stops counting when it detects the falling edge.
2. Drives the level on the SCLn line low.
3. Starts counting the low-level period specified in ICBRL.BRL[4:0].

When the IIC finishes counting the low-level period, it stops driving the SCLn line low to release the line. If the low-level period of the SCL clock signal from the other master device is longer than the low-level period set in the IIC, the low-level period of the SCL signal is extended. When the low-level period for the other master device ends, the SCL signal rises because the SCLn line is released.

When the IIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock rises. That is, when SCL signals from more than one master are contending, the high-level period of the SCL signal is synchronized with that of the clock with the narrower period, and the low-level period of the SCL signal is synchronized with that of the clock with the broader period. However, such synchronization of the SCL signal is only enabled when the SCLE bit in ICFER is set to 1.

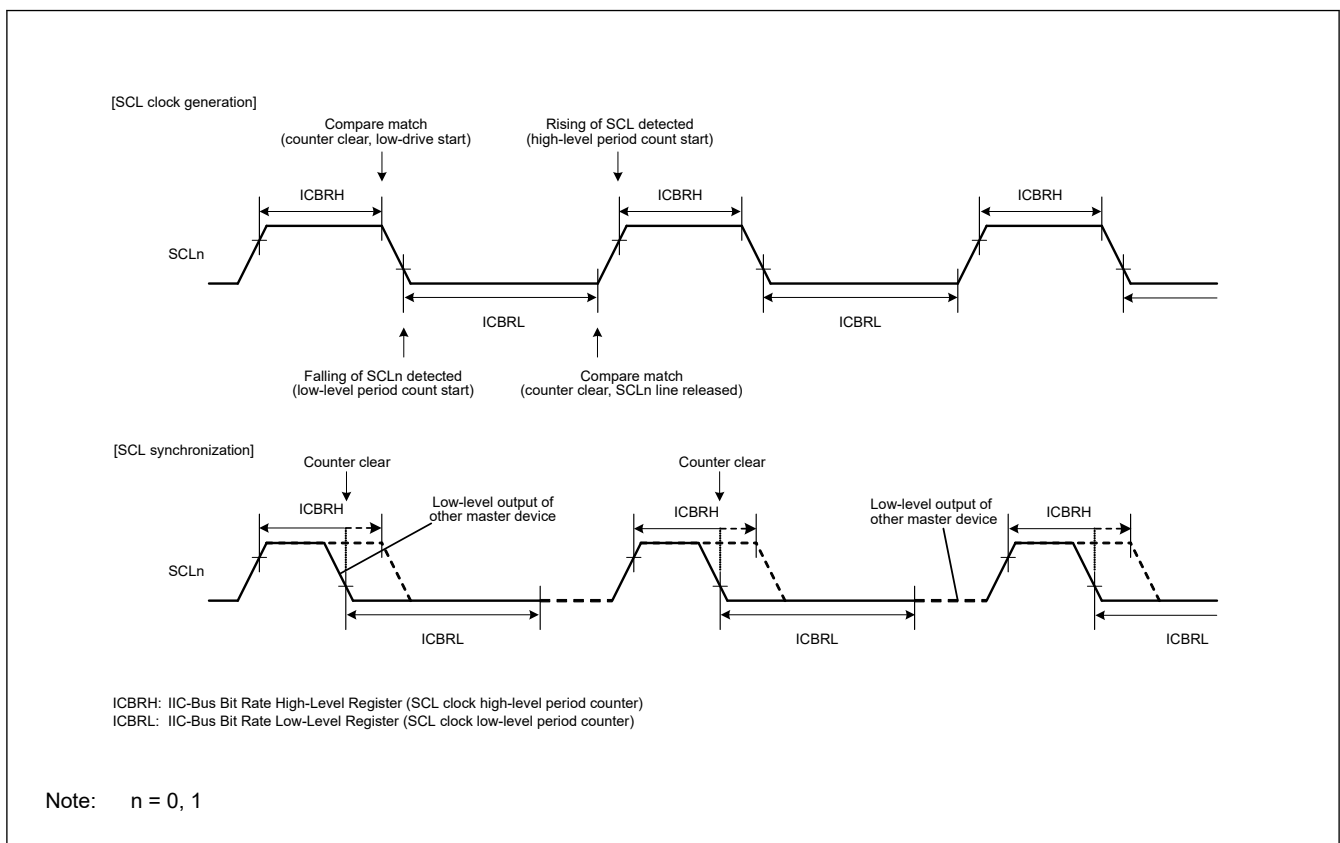


Figure 31.21 Generation and synchronization of SCL signal from IIC

### 31.5 SDA Output Delay Function

The IIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output on the SDA line, including issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals.

With this function, SDA output is delayed from the detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval during which the SCL clock is low. This approach helps prevent erroneous operation of communications devices, with the aim of satisfying the 300-ns minimum data-hold time requirement of the SMBus specification. The output delay function is enabled by setting the SDDL[2:0] bits in ICMR2 to a value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled, for example, the DLCS bit in ICMR2 selects the clock source for the SDA output delay counter, either as the internal base clock (IICφ) for the IIC module or as the internal base clock divided by 2 (IICφ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in ICMR2. When the delay cycles count is reached, the IIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.

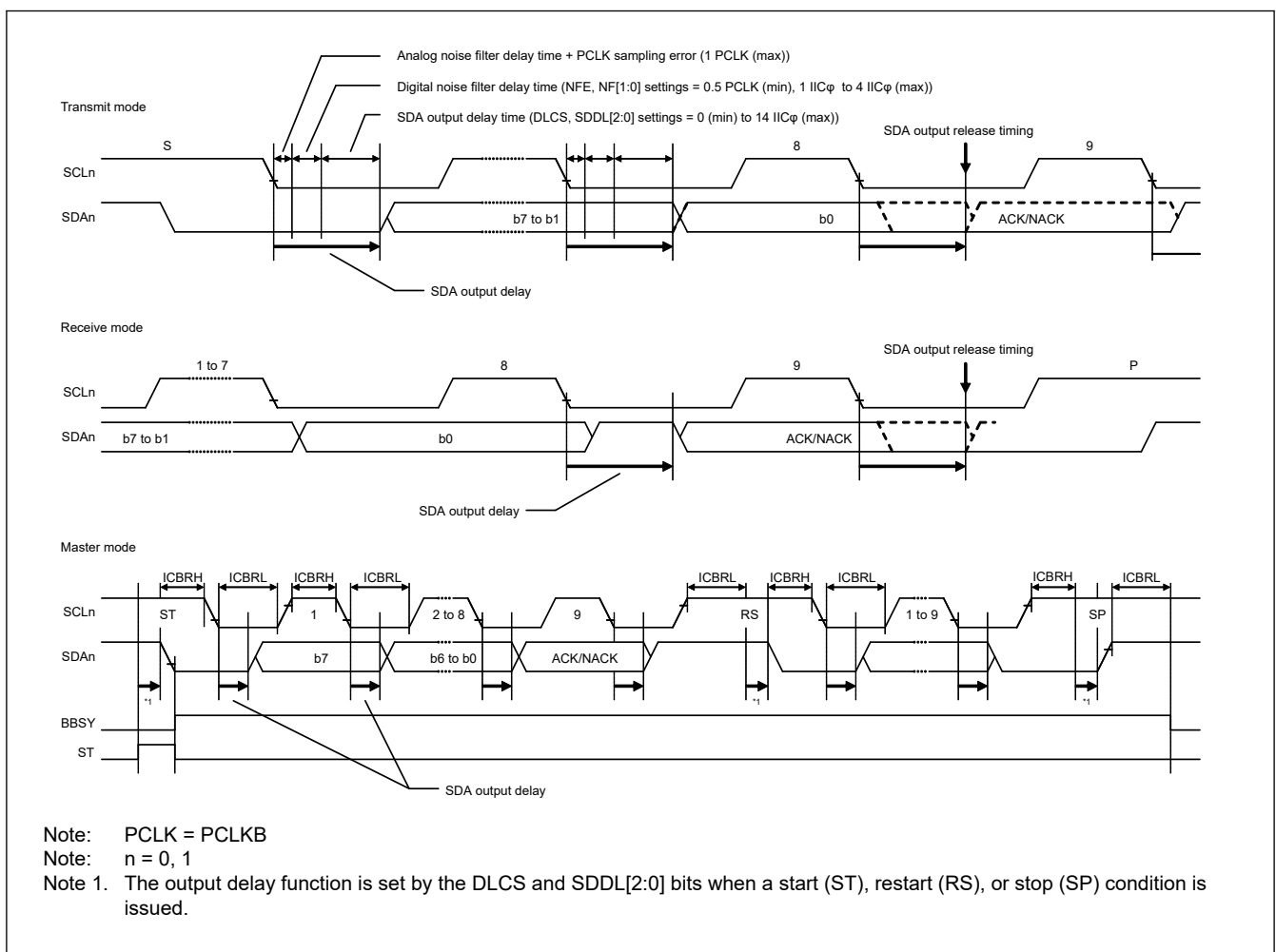


Figure 31.22 SDA output delay function

### 31.6 Digital Noise Filter Circuits

The internal circuitry sees the states of the SCLn and SDAn pins through analog and digital noise-filter circuits. Figure 31.23 shows a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the IIC consists of four flip-flop circuit stages connected in series and a match-detection circuit. The number of valid stages in the digital noise filter is selected in the NF[1:0] bits in ICMR3. The selected number of valid stages determines the noise-filtering capability as a period from 1 to 4 IICφ cycles.

The input signal to the SCL<sub>n</sub> pin (or SDA<sub>n</sub> pin) is sampled on falling edges of the IIC $\phi$  signal. When the input signal level matches the output level of the number of valid flip-flop circuit stages as selected in the NF[1:0] bits in ICMR3, the signal level is seen in the subsequent stage. If the signal levels do not match, the previous value is saved.

If the ratio between the frequency of the internal operating clock (PCLKB) and the transfer rate is small, for example, if data transfer is 400 kbps with PCLKB = 4 MHz, the characteristics of the digital noise filter might lead to the elimination of required signals as noise. In such cases, it is possible to disable the digital noise-filter circuit by setting the ICFER.NFE bit to 0, and use only the analog noise filter circuit.

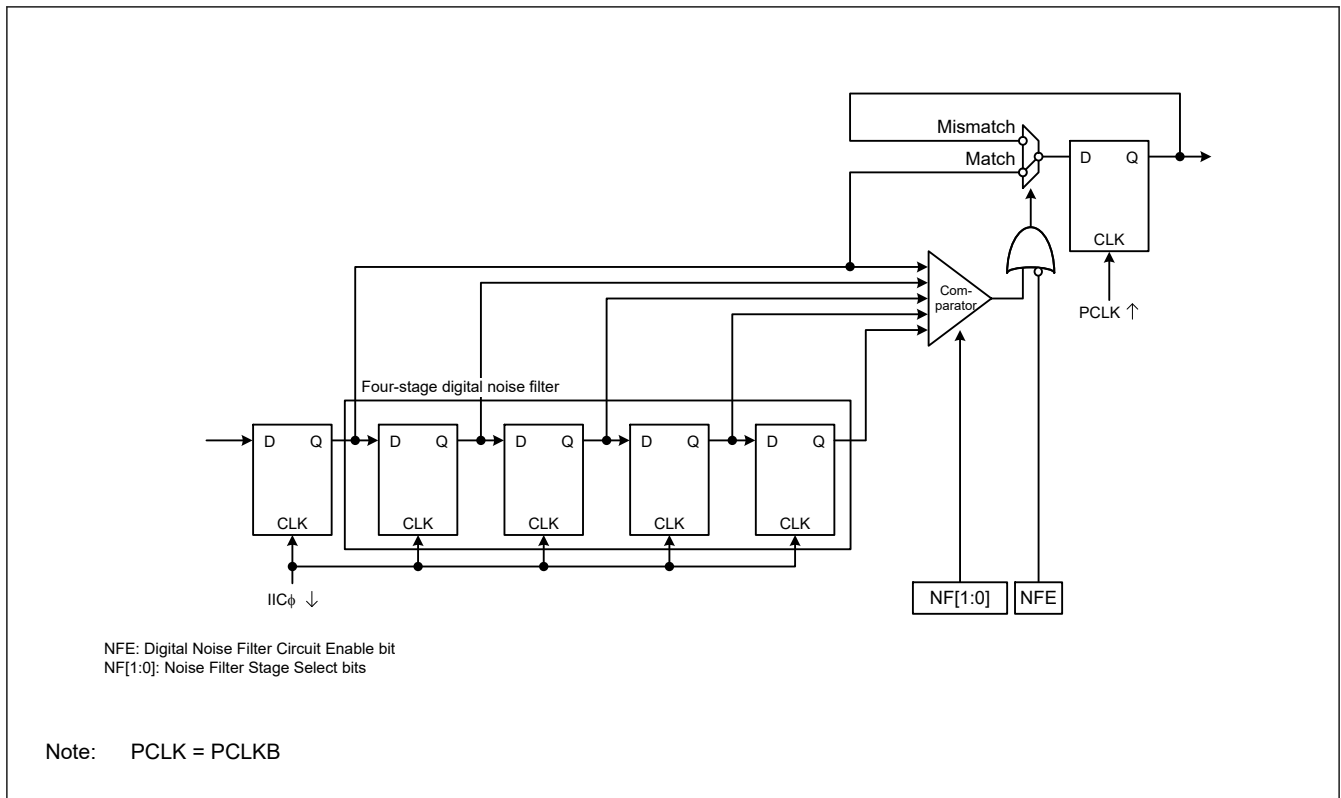


Figure 31.23 Digital noise filter circuit block diagram

## 31.7 Address Match Detection

The IIC can set three unique slave addresses in addition to the general call address and host address. The slave addresses can be 7-bit or 10-bit slave addresses.

### 31.7.1 Slave-Address Match Detection

The IIC can set three unique slave addresses and has a slave address detection function for each unique slave address. When the SAR<sub>y</sub>E bit ( $y = 0$  to 2) in ICSER is set to 1, the slave addresses set in SAR<sub>U</sub><sub>y</sub> and SAR<sub>L</sub><sub>y</sub> ( $y = 0$  to 2) can be detected.

When the IIC detects a match of the set slave address, the associated AAS<sub>y</sub> flag ( $y = 0$  to 2) in ICSR1 is set to 1 on the rising edge of the ninth SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the subsequent R/W# bit. This causes a receive data full interrupt (IIC<sub>n</sub>\_RXI) or transmit data empty interrupt (IIC<sub>n</sub>\_TXI) to be generated. The AAS<sub>y</sub> flag identifies which slave address is specified.

Figure 31.24 to Figure 31.26 show the AAS<sub>y</sub> flag set timing in three cases.

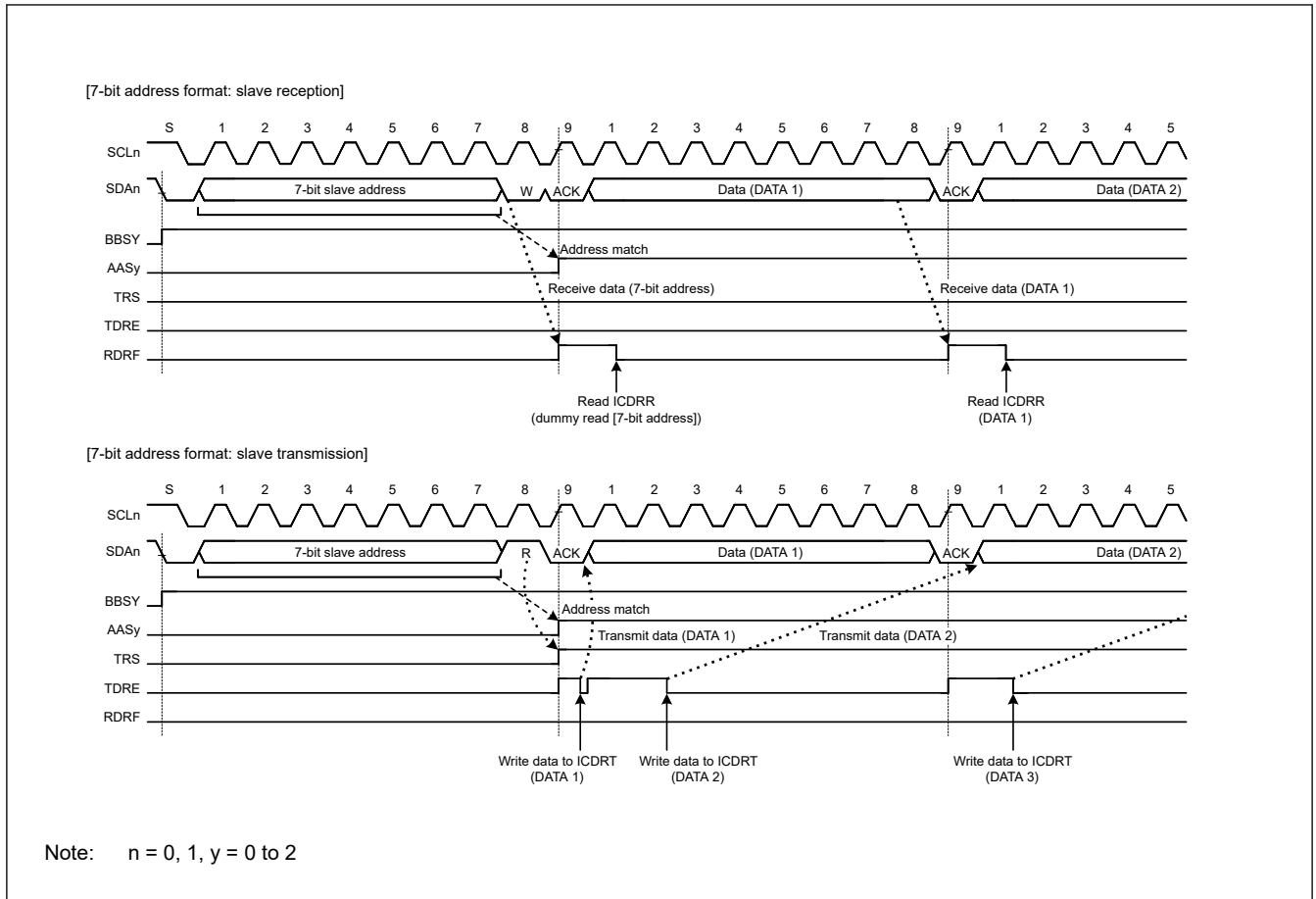


Figure 31.24 AASy flag set timing with 7-bit address format

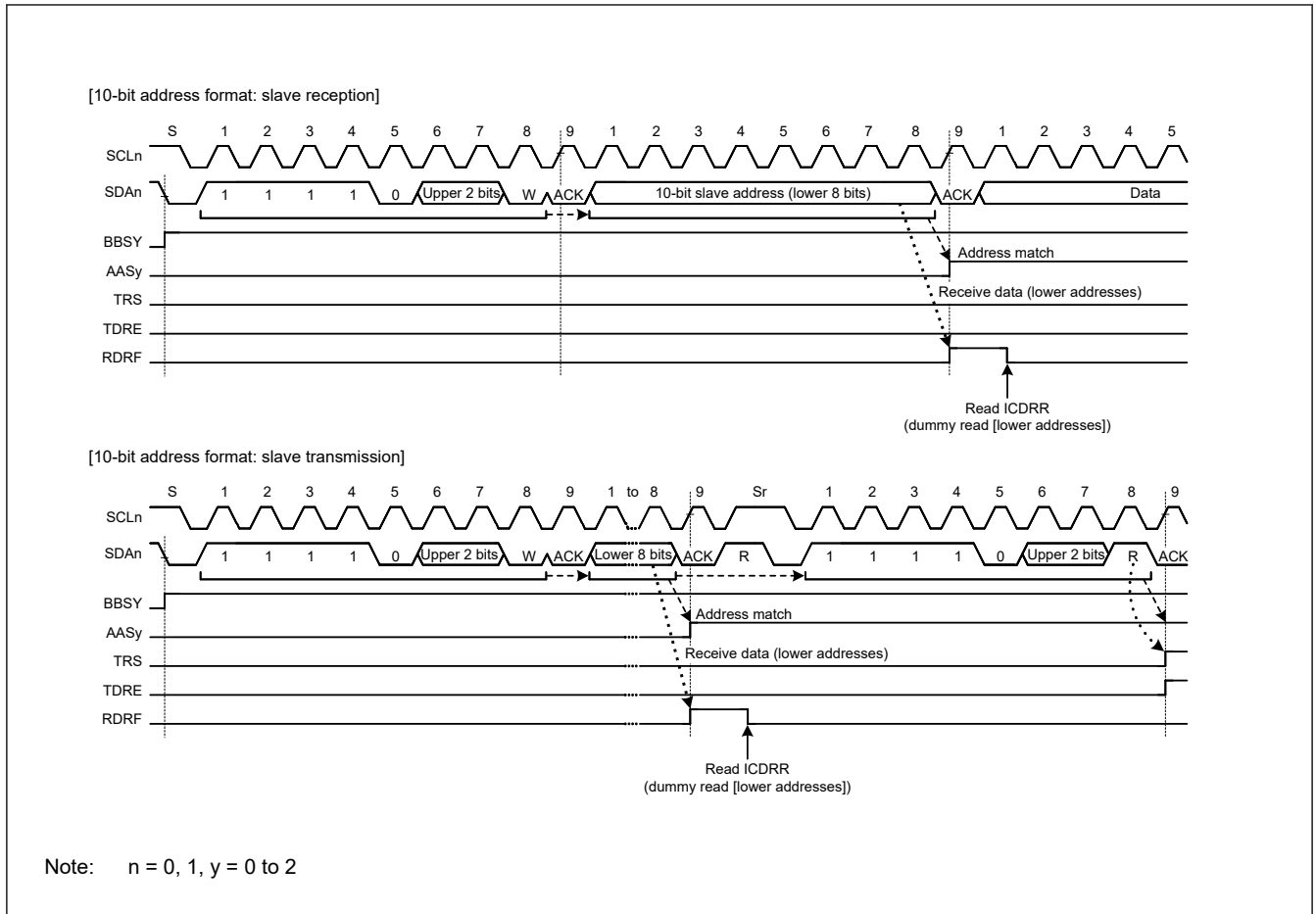


Figure 31.25 AASy flag set timing with 10-bit address format

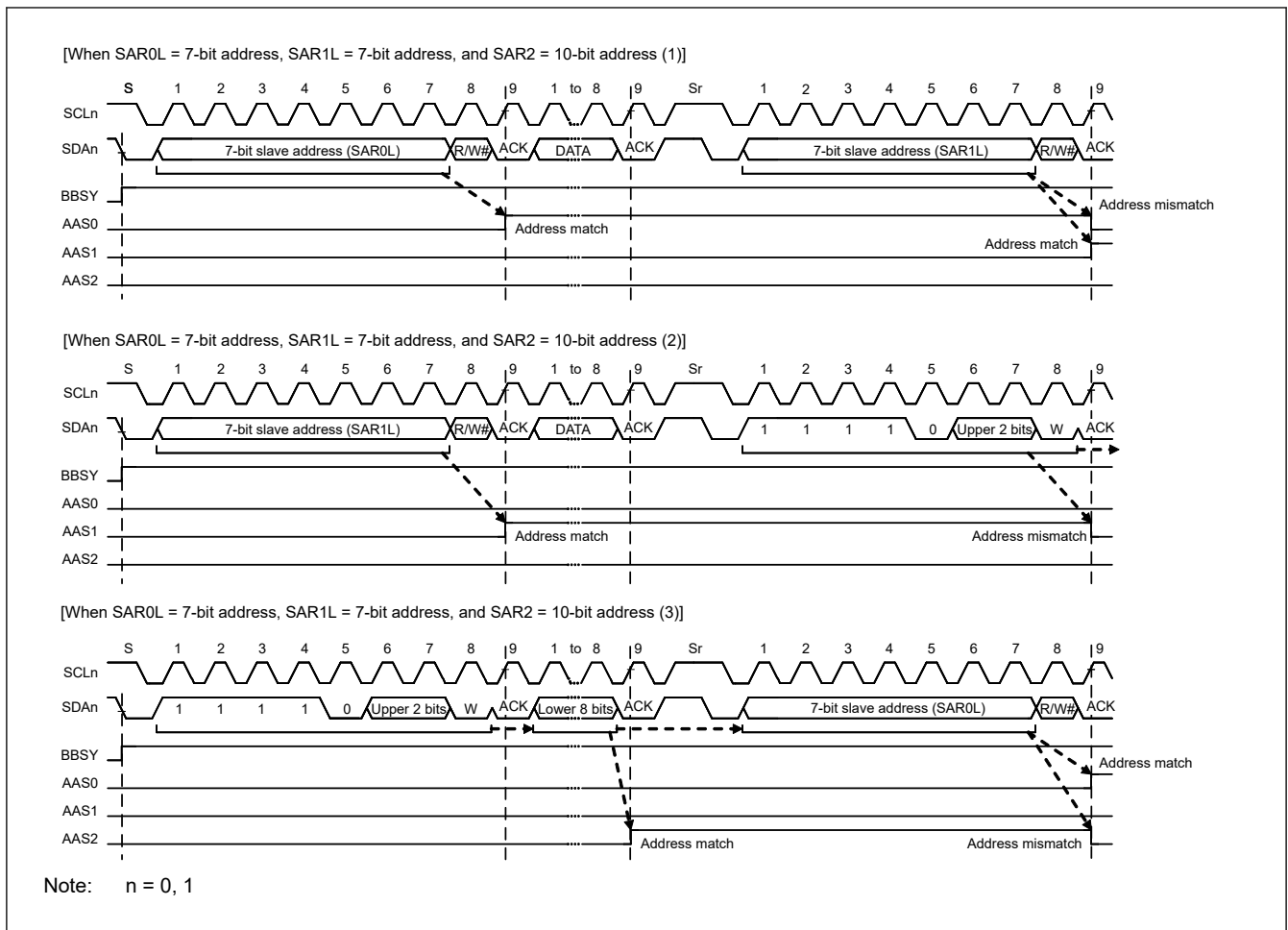


Figure 31.26 AASy flag set and clear timing with mixed 7-bit and 10-bit address formats

### 31.7.2 Detection of General Call Address

The IIC provides detection of the general call address (0000 000b + 0 [W]). This is enabled by setting the GCAE bit in IC SER to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1[R] (start byte), the IIC recognizes this as the address of a slave device with an all-zero address, but not as the general call address.

When the IIC detects the general call address, both the GCA flag in ICSR1 and the RDRF flag in ICSR2 set to 1 on the rising edge of the ninth cycle of the SCL clock. This leads to the generation of a receive data full interrupt (IICn\_RXI). The value of the GCA flag can be checked to confirm that the general call address was transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

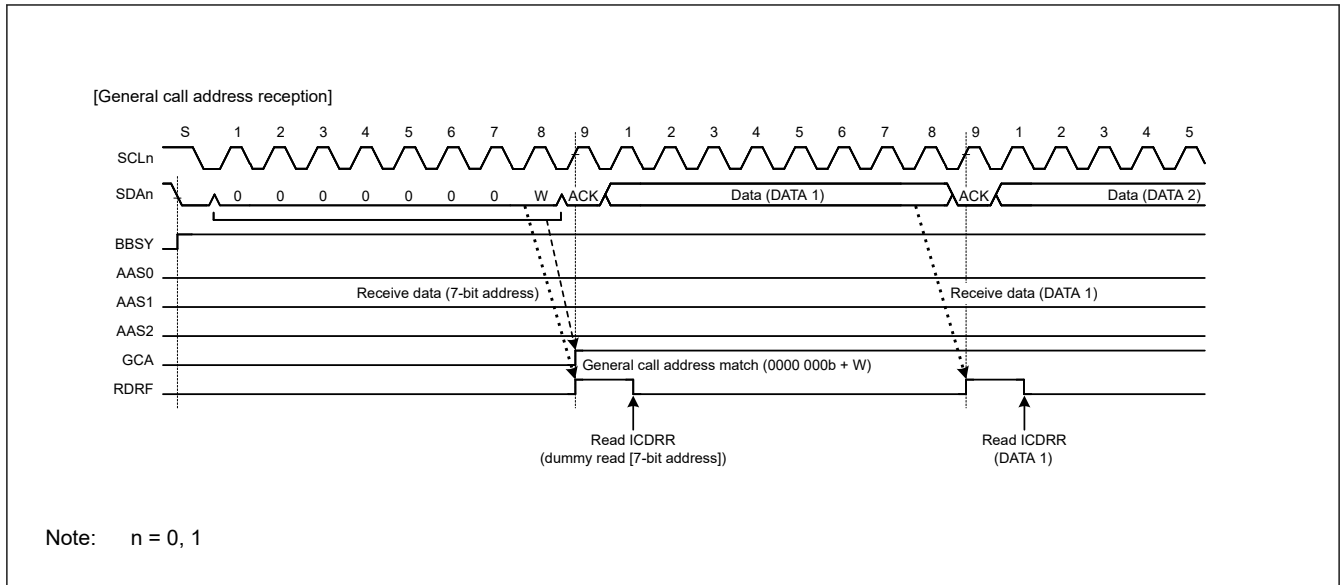


Figure 31.27 Timing of GCA flag setting during reception of general call address

### 31.7.3 Device-ID Address Detection

The IIC module provides detection of device-ID address compliant with the I<sup>2</sup>C bus specification (revision 03). When the IIC receives 1111 100b as the first byte after a start or restart condition is issued with the DIDE bit in ICSER set to 1, it recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the ninth SCL clock cycle when the subsequent R/W# bit is 0, then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the IIC sets the associated AASy flag (y = 0 to 2) in ICSR1 to 1.

When the first byte received after the issue of a start or restart condition matches the device ID address (1111 100b) again and the subsequent R/W# bit is 1, the IIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

In the device ID address detection function, the IIC sets the DID flag to 0 if a match with the IIC slave address is not obtained or a match with the device ID address is not obtained after a match with the IIC slave address and a restart condition is not detected. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b), and the R/W# bit is 0, the IIC sets the DID flag to 1 and compares the second and subsequent bytes with the slave address of the IIC. If the R/W# bit is 1, the DID flag holds the previous value and the IIC does not compare the second and subsequent bytes. Therefore, the reception of a device ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Additionally, prepare the device ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal transmit data. For details on the information that must be included in device ID fields, contact NXP Semiconductors.



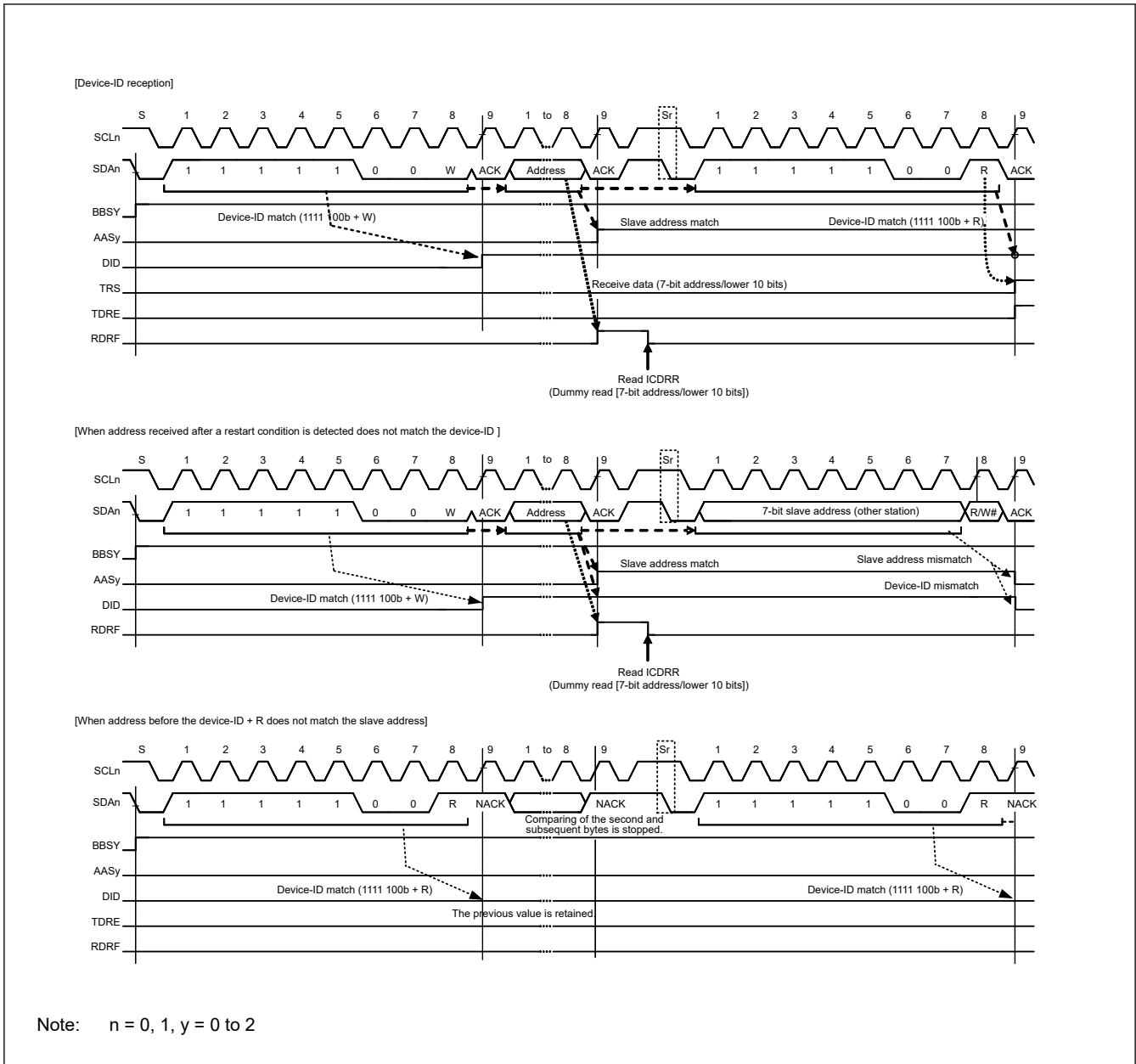


Figure 31.28 AASy and DID flag set and clear timing during reception of device ID

### 31.7.4 Host Address Detection

The IIC provides host address detection when operating in SMBus. When the HOAE bit in ICSER is set to 1 while the SMBS bit in ICMR3 is 1, the IIC can detect the host address (0001 000b) in slave receive mode (MST and TRS bits = 00b in ICCR2).

When the IIC detects the host address, the HOA flag in ICSR1 is set to 1 on the rising edge of the 9th SCL clock cycle, and at the same time, the RDRF flag in ICSR2 is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (IICn\_RXI) to be generated. The HOA flag indicates that the host address was sent from another device.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit = 1), the IIC can also detect the host address. After the host address is detected, the IIC operates in the same manner as in normal slave operation.

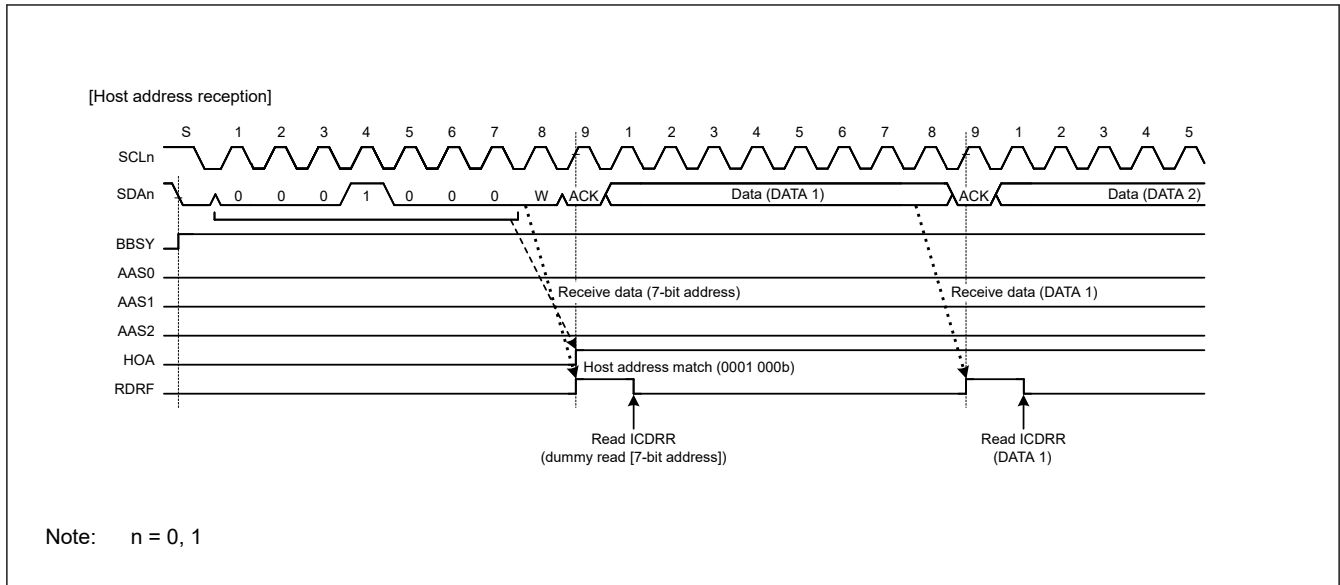


Figure 31.29 HOA flag set timing during reception of host address

### 31.8 Wakeup Function

The IIC provides a wakeup function that causes the MCU to transition from Software Standby mode to normal operation. The wakeup function enables the reception of data when the peripheral module clock (PCLKB) is stopped, and generates a wakeup interrupt signal on a match of the slave address of the received data. This wakeup interrupt signal triggers the return to normal operation. After the wakeup interrupt occurs, switch the IIC to PCLKB synchronous operation so that communication can continue.

The wakeup function has four operation modes:

- Normal wakeup mode 1
- Normal wakeup mode 2
- Command recovery mode
- EEP response mode

Table 31.9 describes the behavior in these modes.

Table 31.9 Wakeup operation modes

Operation mode	ACK response timing	ACK response before wakeup to PCLKB synchronous operation	SCL state during wakeup to PCLKB synchronous operation
Normal wakeup mode 1	Before wakeup to PCLKB synchronous operation <sup>*1</sup>	ACK	Fixed low
Normal wakeup mode 2	After wakeup to PCLKB synchronous operation <sup>*2</sup>	Before wakeup: no response (NACK level retained) After wakeup: ACK response	Fixed low
Command recovery mode	Before wakeup to PCLKB synchronous operation <sup>*1</sup>	ACK	Open
EEP response mode	Before recovery to PCLKB synchronous operation <sup>*1</sup>	NACK	Open

Note 1. Switching timing from PCLKB asynchronous operation to PCLKB synchronous operation is the falling edge of the 9th clock of the SCL.

Note 2. Switching timing from PCLKB asynchronous operation to PCLKB synchronous operation is the falling edge of the 8th clock of the SCL.

The following can be selected as wakeup interrupt sources:

- Host address detection (valid when IC SER.HOAE = 1)
- General call address detection (valid when IC SER.GCAE = 1)

- Slave address 0\*1 detection (valid when IC SER.SAR0E = 1)
- Slave address 1\*1 detection (valid when IC SER.SAR1E = 1)
- Slave address 2\*1 detection (valid when IC SER.SAR2E = 1)

Note 1. Only 7-bit address can be set. Set the FS bit in SARU<sub>y</sub> (y = 0 to 2) to 0.

### Precautions on the use of the wakeup function

- Do not change the content of the IIC registers except the IC IER register and WUSEN bit in ICWUR2 while the WUASYF flag in ICWUR2 is 1 (during PCLKB asynchronous operation).
- Set ICWUR.WUE and ICWUR.WUIE to 1, and ICCR2.MST and ICCR2.TRS to 0 (slave reception mode) before switching to PCLKB asynchronous mode.
- The device ID and the 10-bit slave address cannot be selected for the wakeup interrupt source. Set the DIDE bit in IC SER and FS bit in SARU<sub>y</sub> (y = 0 to 2) to 0.
- Set bits TIE, TEIE, RIE, NAKIE, SPIE, STIE, ALIE, and TMOIE in the IC IER register to 0 (interrupt disabled) after switching to PCLKB asynchronous operation (ICWUR2.WUASYF = 1).
- When the wakeup function is enabled, do not use the timeout function (ICWUR.WUE = 1)
- Even when a wakeup interrupt is generated during PCLKB asynchronous operation (when ICWUR2.WUASYF = 1), if the slave addresses match in PCLKB synchronous mode (ICWUR2.WUASYF = 0), the wakeup interrupt does not occur and the WUF flag is not set.
- If the timing of writing 0 to the ICWUR2.WUSEN bit and the timing of detecting a start condition conflict, the IIC might start the next reception in PCLKB synchronous operation mode. In this case, ICWUR2.WUASYF flag becomes 1 (switch to PCLKB asynchronous mode) when data communication is complete, a stop condition is detected, and detection of a wakeup event starts.
- If you want to switch from PCLKB asynchronous operation to PCLKB synchronous operation without address match detection, it will switch in the stop condition detection. When the ICWUR2.WUSEN bit was set to 1 in a bus free state, it is continued PCLKB asynchronous operation (Reception operation: waiting communication frame). ICWUR2.WUASYF flag becomes to 1 when IIC detect the stop condition of the next communication frame, and IIC switches to PCLKB synchronous operation.
- After writing 0 to the WUSEN bit in ICWUR2, do not change registers relate to the IIC operation mode setting (ICMR3, IC SER, and SARL<sub>y</sub>) until the mode is switched to PCLKB asynchronous operation from PCLKB synchronous operation (while the ICWUR2.WUASYF flag is 1). If the register value changes during this period by an interrupt handling or another factor, the IIC might malfunction before switching to the asynchronous operation.
- During PCLKB asynchronous operation (ICWUR2.WUASYF = 0 (or WUASYF = 1)), do not refer to each flag of IC SR1, IC SR2 register and ICCR2.BBSY flag.

#### 31.8.1 Normal Wakeup Mode 1

This section describes the behavior, the timing, and an example operation of normal wakeup mode 1.

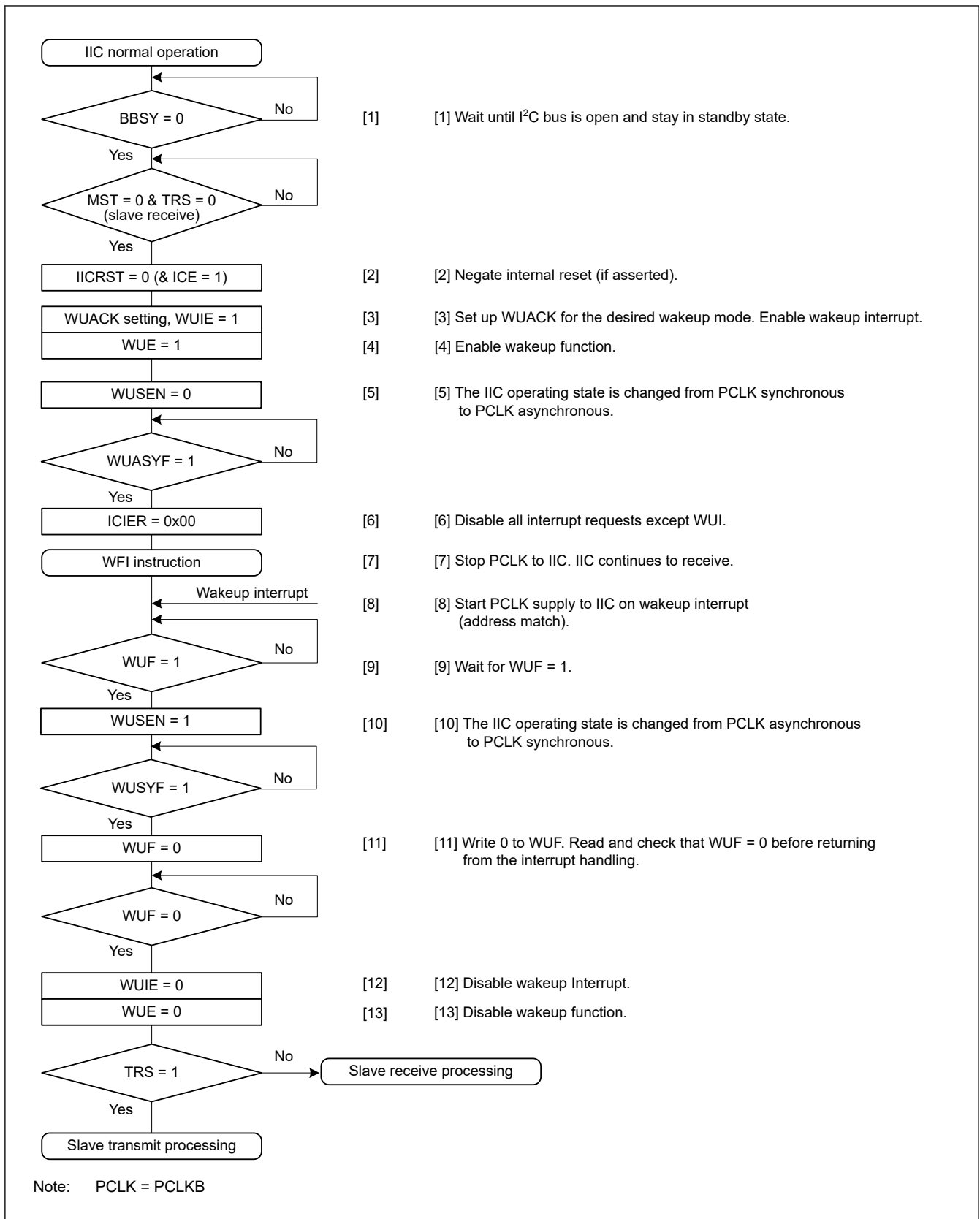
In normal wakeup mode 1, a wakeup interrupt triggered by the match of the slave address initiates the transition to normal operation as follows:

- Before wakeup: ACK is sent in response to the data received with its own slave address of the IIC.
- During wakeup: ACK response is made on the 9th clock cycle of SCL, after which SCL is held low\*1.
- After wakeup: Normal operation continues.

Note 1. Between the 9th clock cycle and 1st clock cycle during wakeup, ICMR1.WAIT = 1 is invalid.

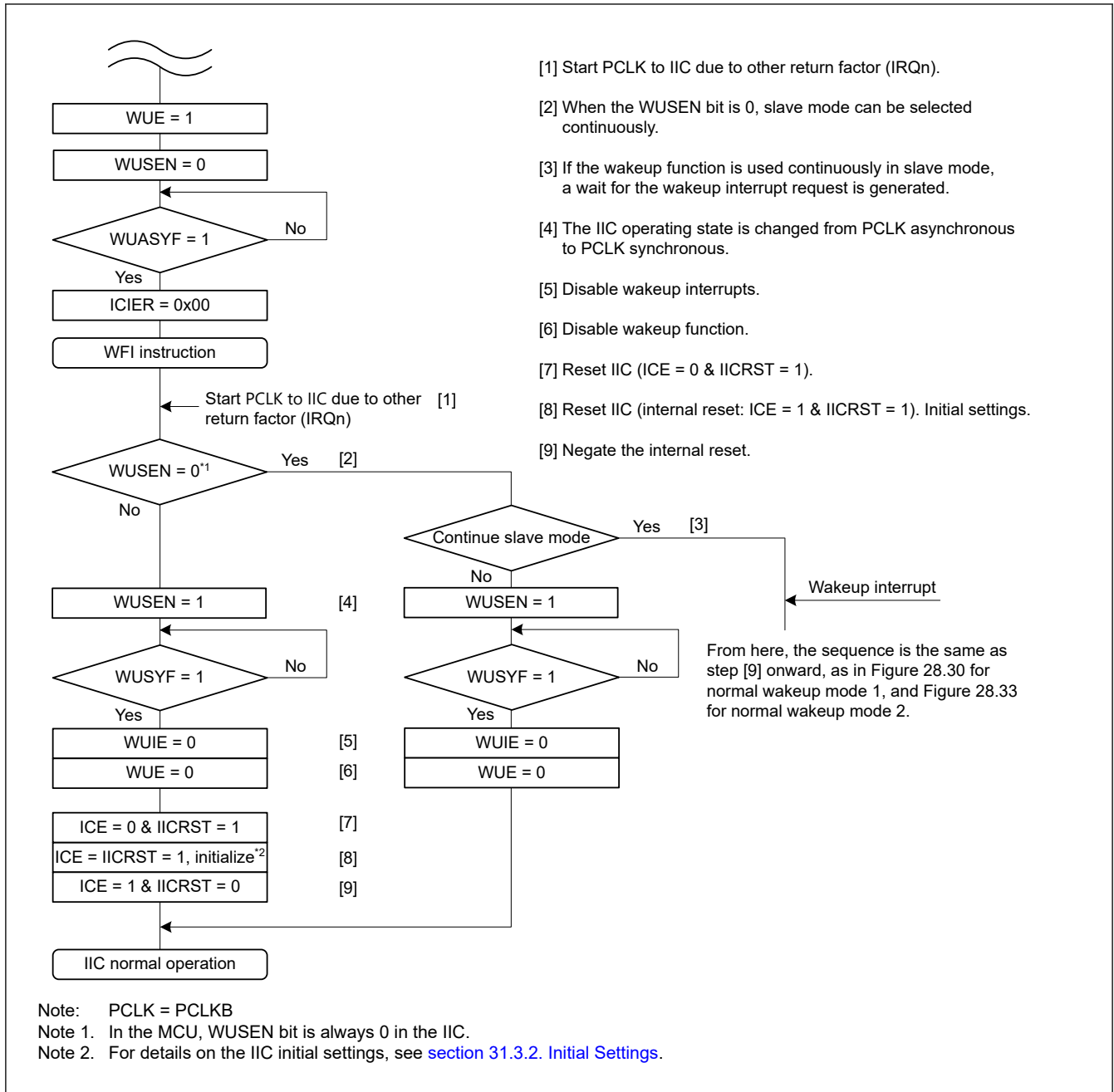
If the slave address does not match, the SCL line is not held low after the 9th clock cycle of SCL, and the slave operation continues. [Figure 31.30](#) shows an operation example, and [Figure 31.32](#) shows the detailed timing.

If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, for example the IRQn, the WUF flag is not set to 1. [Figure 31.31](#) shows an operation example.



**Figure 31.30 Example operation of normal wakeup mode 1 when wakeup is triggered by a wakeup interrupt on match of the slave address**

Note: See [Precautions on the use of the wakeup function](#).



**Figure 31.31 Example operation of normal wakeup modes 1 and 2 when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, the IRQn**

Note: For details on the IIC initial settings, see section 31.3.2. Initial Settings.

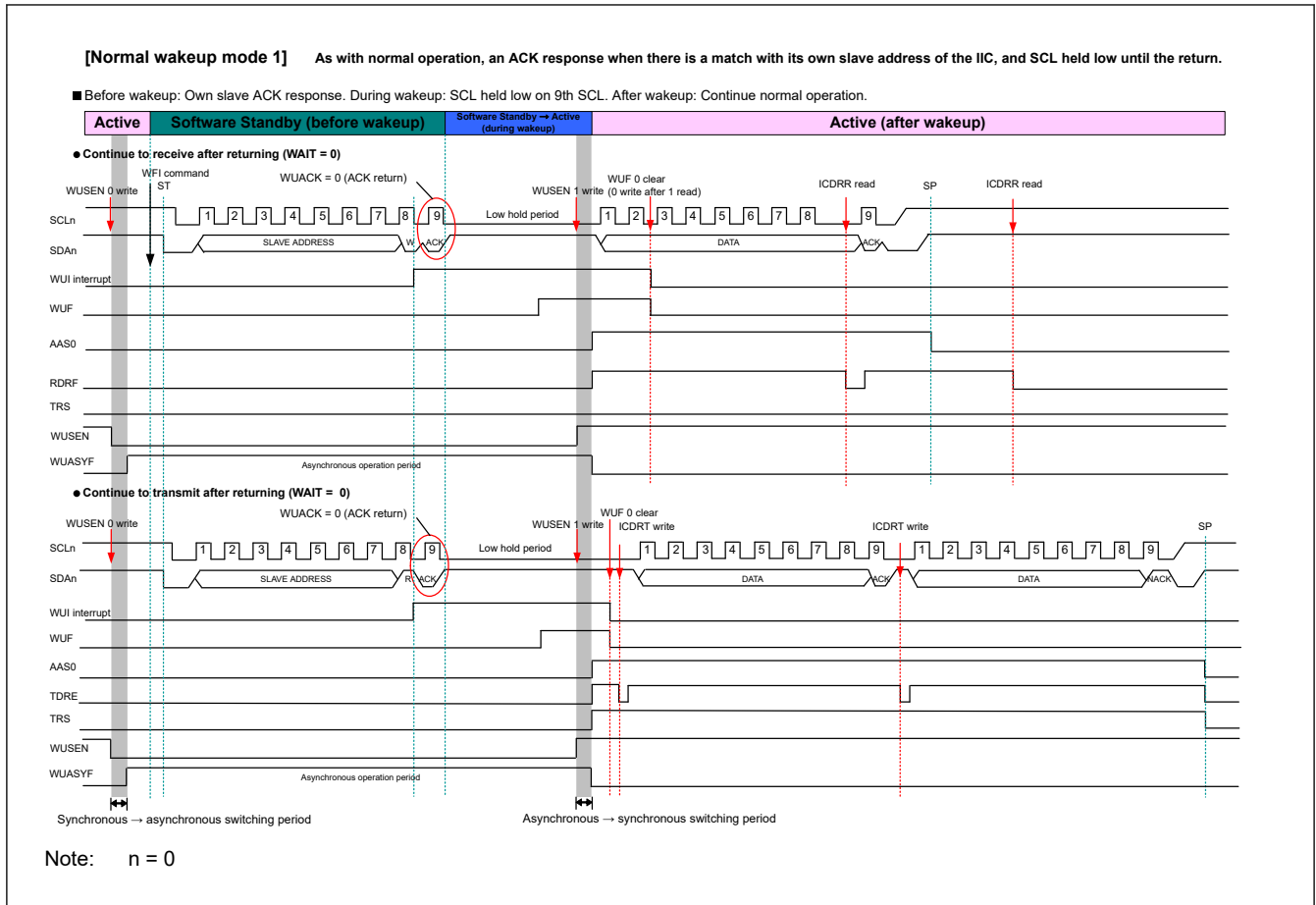


Figure 31.32 Timing of normal wakeup mode 1

### 31.8.2 Normal Wakeup Mode 2

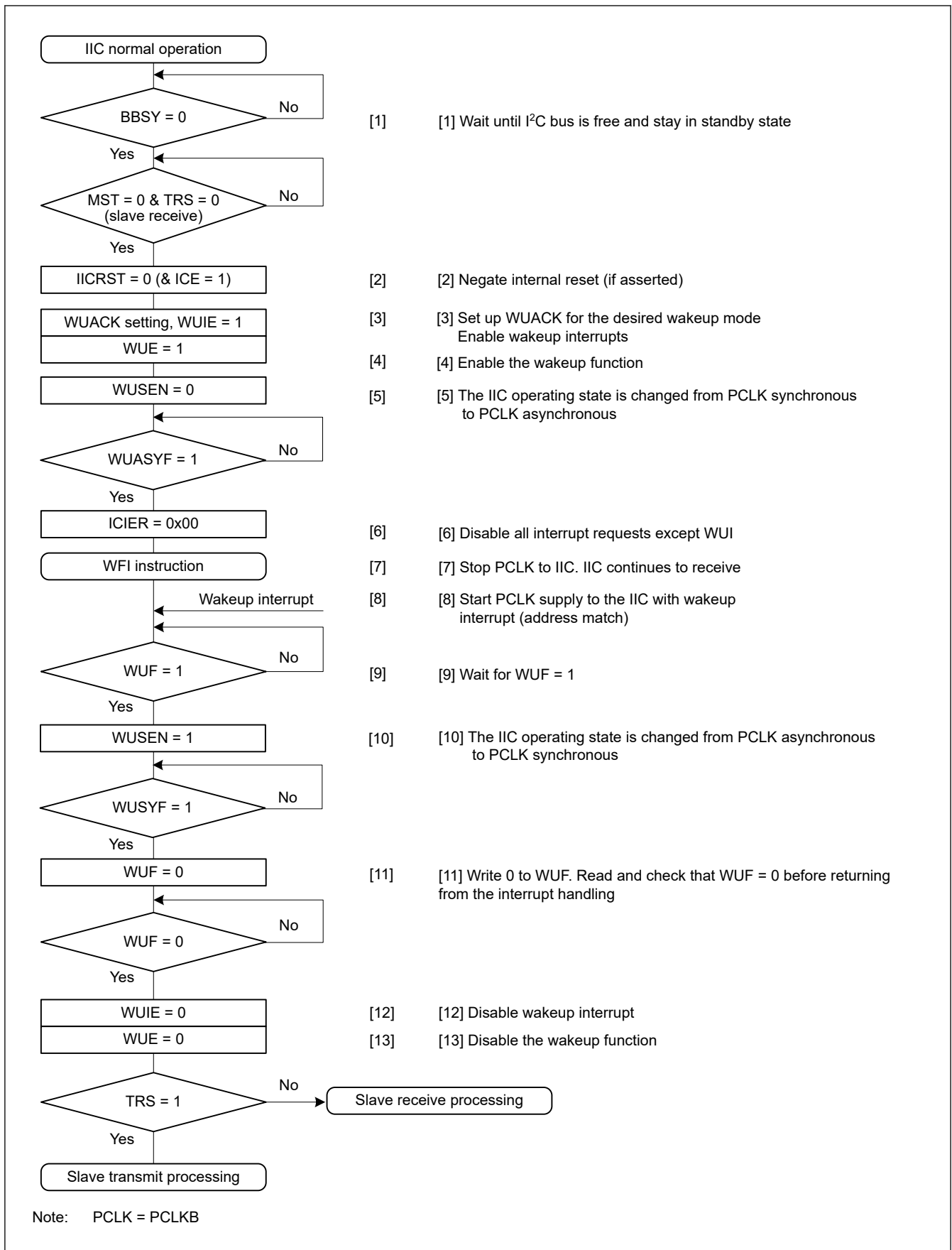
This section describes the behavior, the timing, and an example operation of normal wakeup mode 2.

In normal wakeup mode 2, a wakeup interrupt triggered by a match of the slave address initiates the transition to normal operation as follows:

- Before wakeup: No response to data received with its own slave address until the end of the 8th SCL cycle.
- During wakeup: SCL line held low during the 8th and 9th clock cycles.
- After wakeup: ACK returns on the 9th clock cycle of SCL, and normal operation continues.

If the slave address does not match, the SCL line is not held low after the 8th SCL clock cycle, and the slave operation continues. Figure 31.33 shows an example operation, and Figure 31.34 shows the detailed timing.

If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, such as the IRQn, for example, the WUF flag is not set to 1. Figure 31.31 shows an operation example.



**Figure 31.33 Example operation of normal wakeup mode 2 when wakeup is triggered by a wakeup interrupt on match of the slave address**

Note: See [Precautions on the use of the wakeup function.](#)

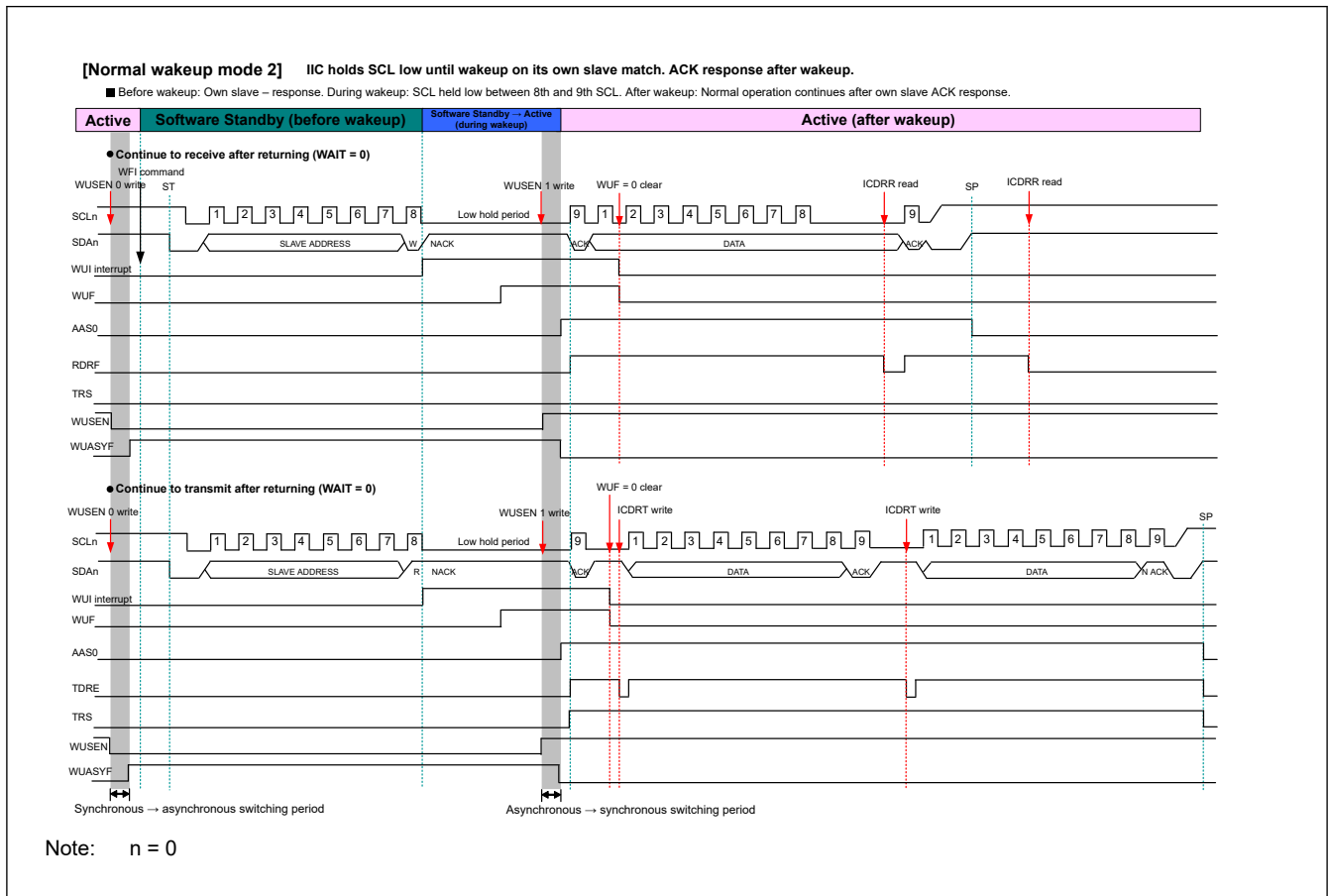


Figure 31.34 Timing of normal wakeup mode 2

### 31.8.3 Command Recovery Mode and EEP Response Mode (Special Wakeup Modes)

This section describes the behavior, the timing, and example operations of the command recovery and EEP response modes. In the command recovery and EEP response modes, the SCL line is not held low during the wakeup period (after the rise of the 9th clock cycle of SCL). Therefore, other I2C devices can use the I2C bus during this period.

A wakeup interrupt triggered by a match of the slave address initiates the transition to normal operation as follows:

- Before wakeup: In response to the data received with its own slave address, the IIC returns ACK (command recovery mode) or NACK (EEP response mode).
- During wakeup: The SCL line is not held low.
- After wakeup: Normal operation continues after IIC initialization.

If the slave address does not match, the slave operation continues.

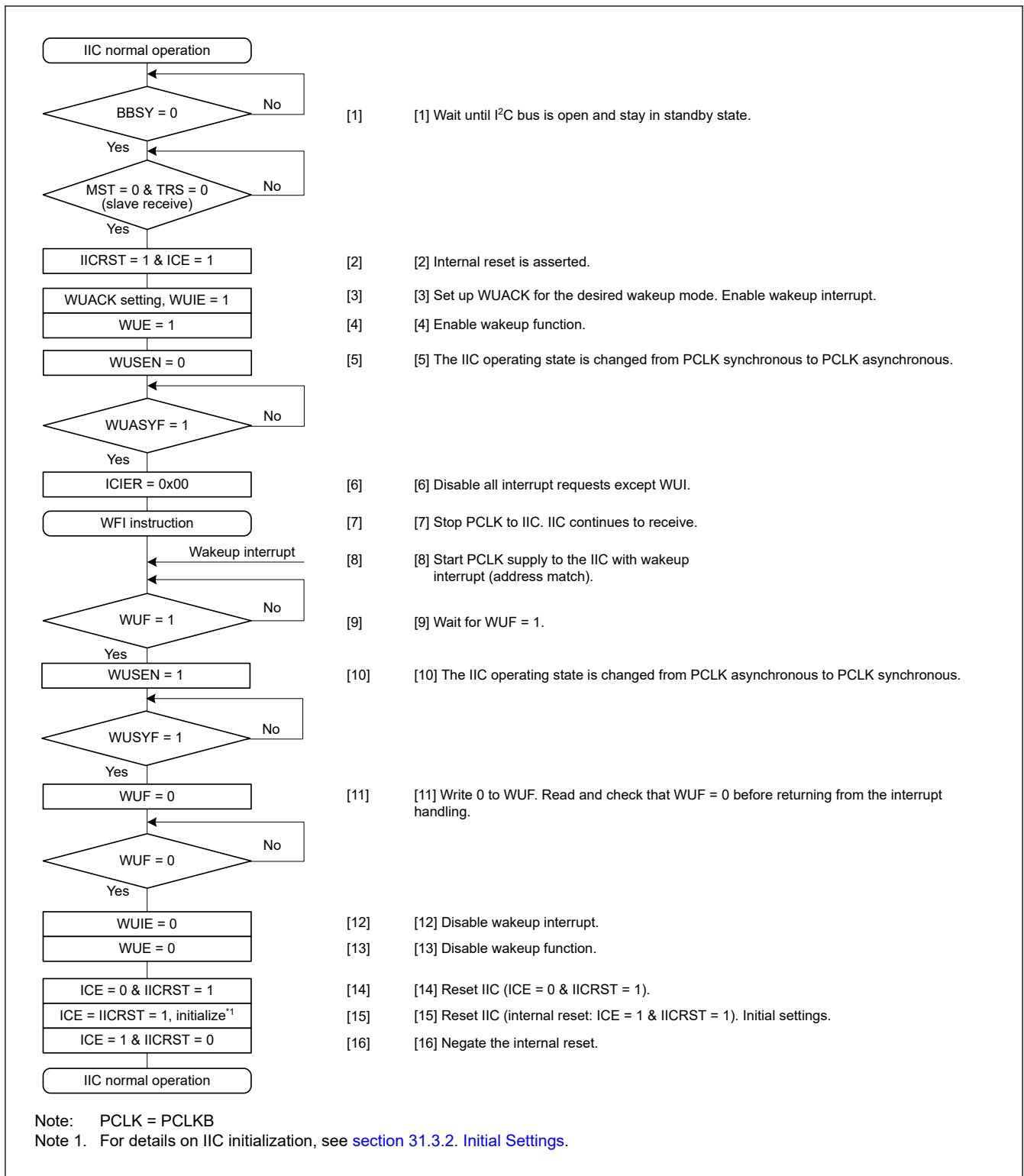
Note: Because the SCL line is not held low during wakeup, transmission or reception of the data that follows the slave address is not possible.

Note: The command recovery and EEP response modes are internal reset states (ICE = IICRST = 1). Therefore, the match of the slave address does not set the flags, HOA, GCA, ASS0, ASS1, and ASS2 in the ICSR1 register.

Figure 31.35 shows an example operation in recovery and EEP response modes. Figure 31.37 shows the detailed timing.

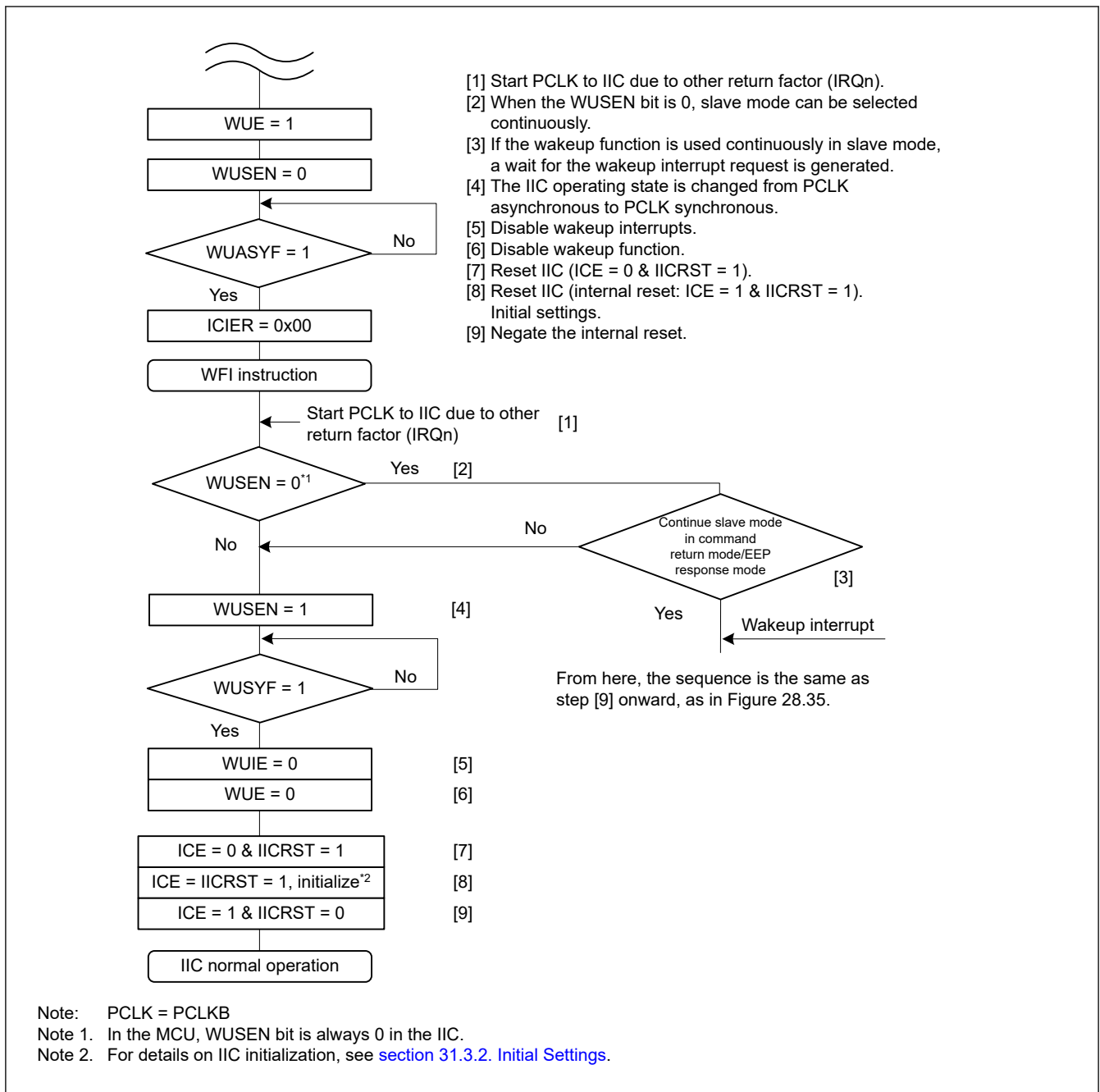
If the transition from Software Standby mode is triggered by an interrupt other than a wakeup interrupt, such as IRQn for example, the WUF flag is not set to 1. Follow the processing shown in Figure 31.36.





**Figure 31.35 Example operation of command recovery mode and EEP response mode when wakeup is triggered by a wakeup interrupt on match of the slave address**

Note: See [Precautions on the use of the wakeup function](#).



**Figure 31.36** Example operation of command recovery and EEP response modes when wakeup is triggered by an interrupt other than IIC wakeup interrupt, for example, the IRQn

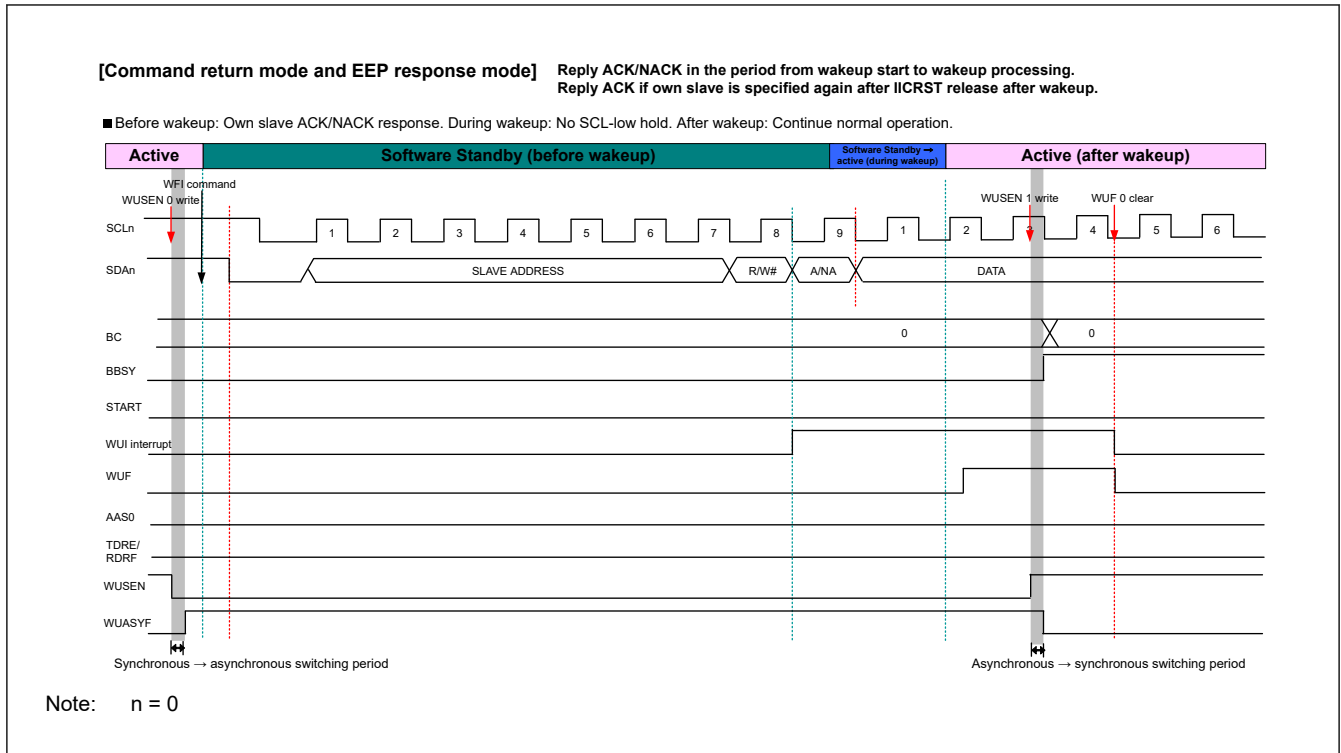


Figure 31.37 Timing of command recovery and EEP response modes

### 31.9 Automatic Low-Hold Function for SCL

#### 31.9.1 Function to Prevent Wrong Transmission of Transmit Data

If the I<sup>2</sup>C Bus Shift Register (ICDRS) is empty and data has not been written to the I<sup>2</sup>C Bus Transmit Data Register (ICDRT) with the IIC in transmission mode (TRS bit = 1 in ICCR2), the SCLn line is automatically held low over the subsequent intervals. This low-hold period is extended until the transmit data is written, which prevents the unintended transmission of erroneous data.

Master transmit mode:

- Low-level interval after a start or restart condition is issued
- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

Slave transmit mode:

- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

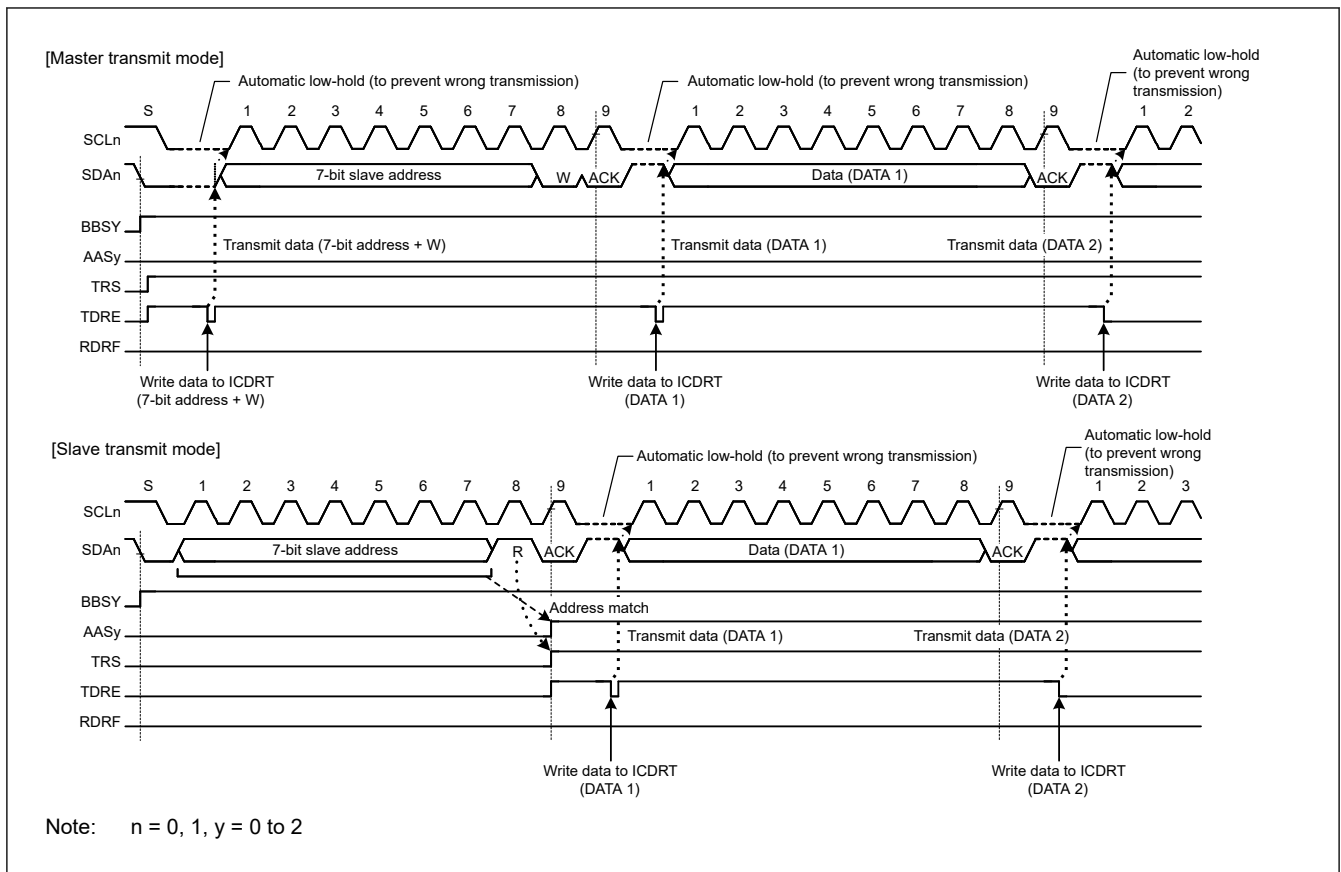


Figure 31.38 Automatic low-hold operation in transmit mode

### 31.9.2 NACK Reception Transfer Suspension Function

This function suspends transfer operation when NACK is received in transmit mode (TRS bit = 1 in ICCR2). This function is enabled when the NACKE bit in ICFER is set to 1. If the next transmit data is already written (TDRE flag = 0 in ICSR2) when NACK is received, the next data transmission on the falling edge of the 9th SCL clock cycle is automatically suspended. This prevents the SDAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF flag = 1 in ICSR2), transmit and receive operations are discontinued. To restore transmit or receive operation, after issuing the restart condition, you need to set the NACKF flag to 0 and try again, or set the NACKF flag to 0 after issuing the stop condition and then start again from issuing the start condition.

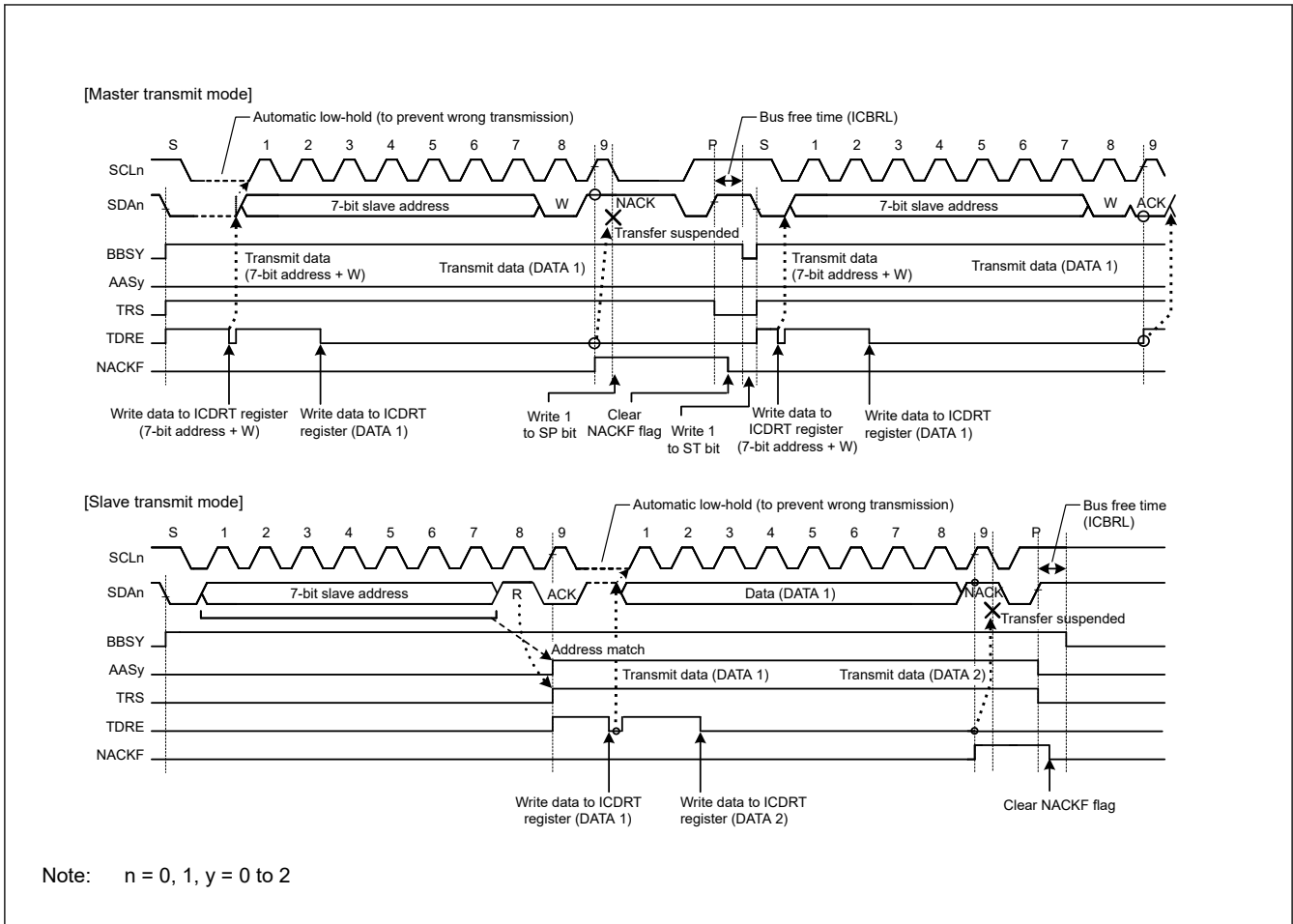


Figure 31.39 Suspension of data transfer when NACK is received, when NACKE = 1

### 31.9.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the IIC holds the SCLn line low automatically immediately before the next data is received to prevent a failure to receive data.

This function is enabled even if the read processing of the final receive data is delayed and, in the meantime, the IIC slave address is designated after a stop condition is issued. This function does not interfere with other communication because the IIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.

Periods in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

#### (1) 1-byte receive operation and automatic low-hold function using the WAIT bit

When the WAIT bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the WAIT bit function. Additionally, when the ICMR3.RDRFS bit is 0, the IIC automatically sends the ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the 8th SCL clock cycle to the falling edge of the 9th SCL clock cycle, and automatically holds the SCLn line low on the falling edge of the 9th SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRR, which enables byte-wise receive operation.

The WAIT bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

#### (2) 1-byte receive operation (ACK/NACK transmission control) and automatic low-hold function using the RDRFS bit

When the RDRFS bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the RDRF flag in ICSR2 is set to 1 (receive data full) on the rising edge of the eighth SCL clock cycle, and the SCLn line is automatically held low on the falling edge of the eighth SCL clock cycle. This low-hold

is released by writing a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, which enables receive operation through the ACK or NACK transmission control based on the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

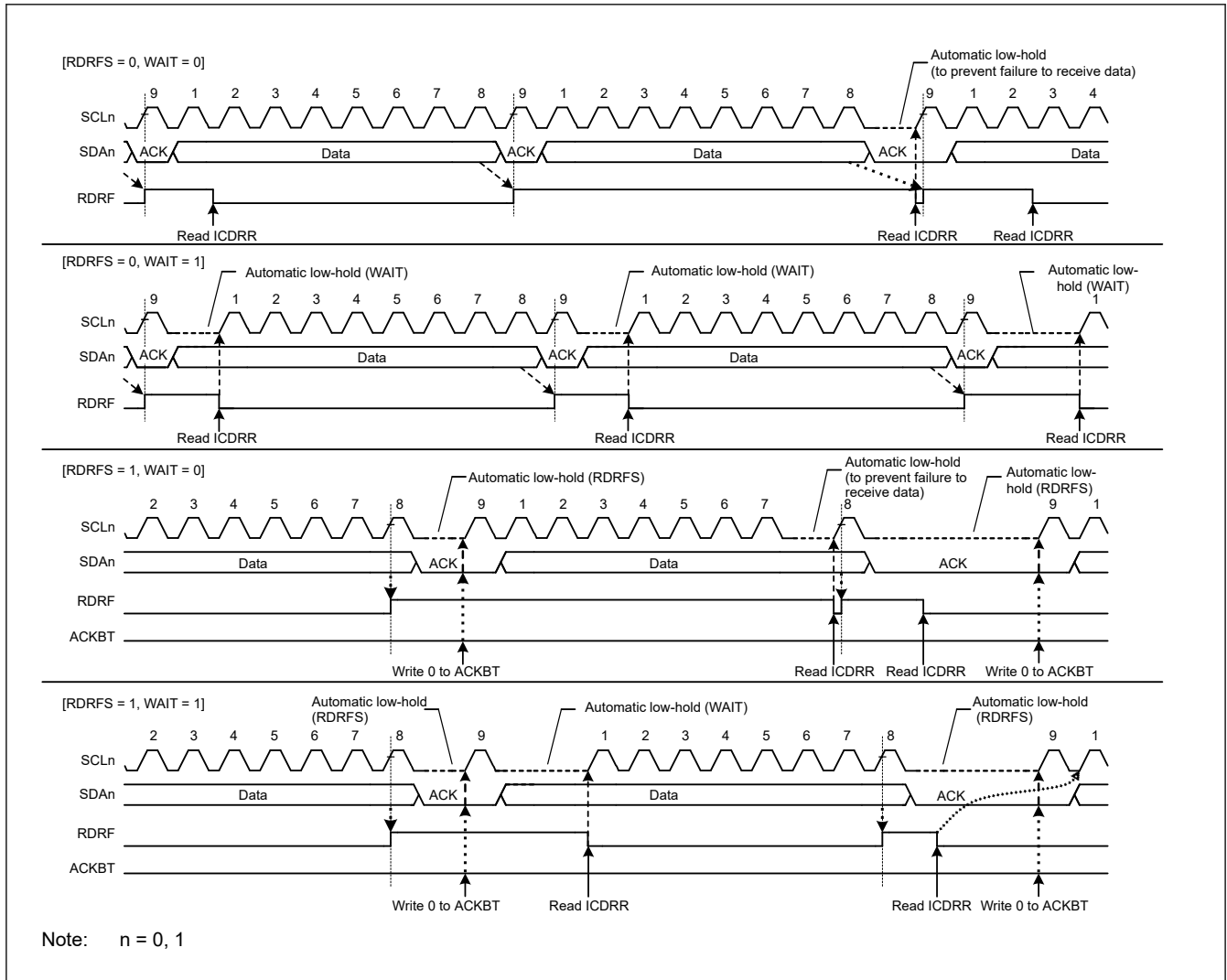


Figure 31.40 Automatic low-hold operation in receive mode using the RDRFS and WAIT bits

### 31.10 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I<sup>2</sup>C bus standard, the IIC provides functions to prevent double-issue of a start condition, detect arbitration-lost during transmission of NACK, and detect arbitration-lost in slave transmit mode.

#### 31.10.1 Master Arbitration-Lost Detection (MALE Bit)

The IIC drives the SDA<sub>n</sub> line low to issue a start condition. However, if the SDA<sub>n</sub> line was already driven low by another master device issuing a start condition, the IIC regards its own start condition as an error and considers this a loss in arbitration. Priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), the IIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration. This prevents a failure of transfer resulting from a start condition being issued while transfer is in progress.

When a start condition is issued successfully, if the transmit data including the address bits (internal SDA output level) and the level on the SDA<sub>n</sub> line do not match (high output as the internal SDA output, meaning the SDA<sub>n</sub> pin is in the high-impedance state and a low level is detected on the SDA<sub>n</sub> line), the IIC loses in arbitration.

After a loss in arbitration of mastership, the IIC immediately enters slave receive mode. If a slave address, including the general call address, matches its own address at this time, the IIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the MALE bit in ICCFER is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Mismatching of the internal level for output on SDA and the level on the SDA<sub>n</sub> line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 is set to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA<sub>n</sub> line in master transmit mode (MST and TRS bits = 11b in ICCR2).

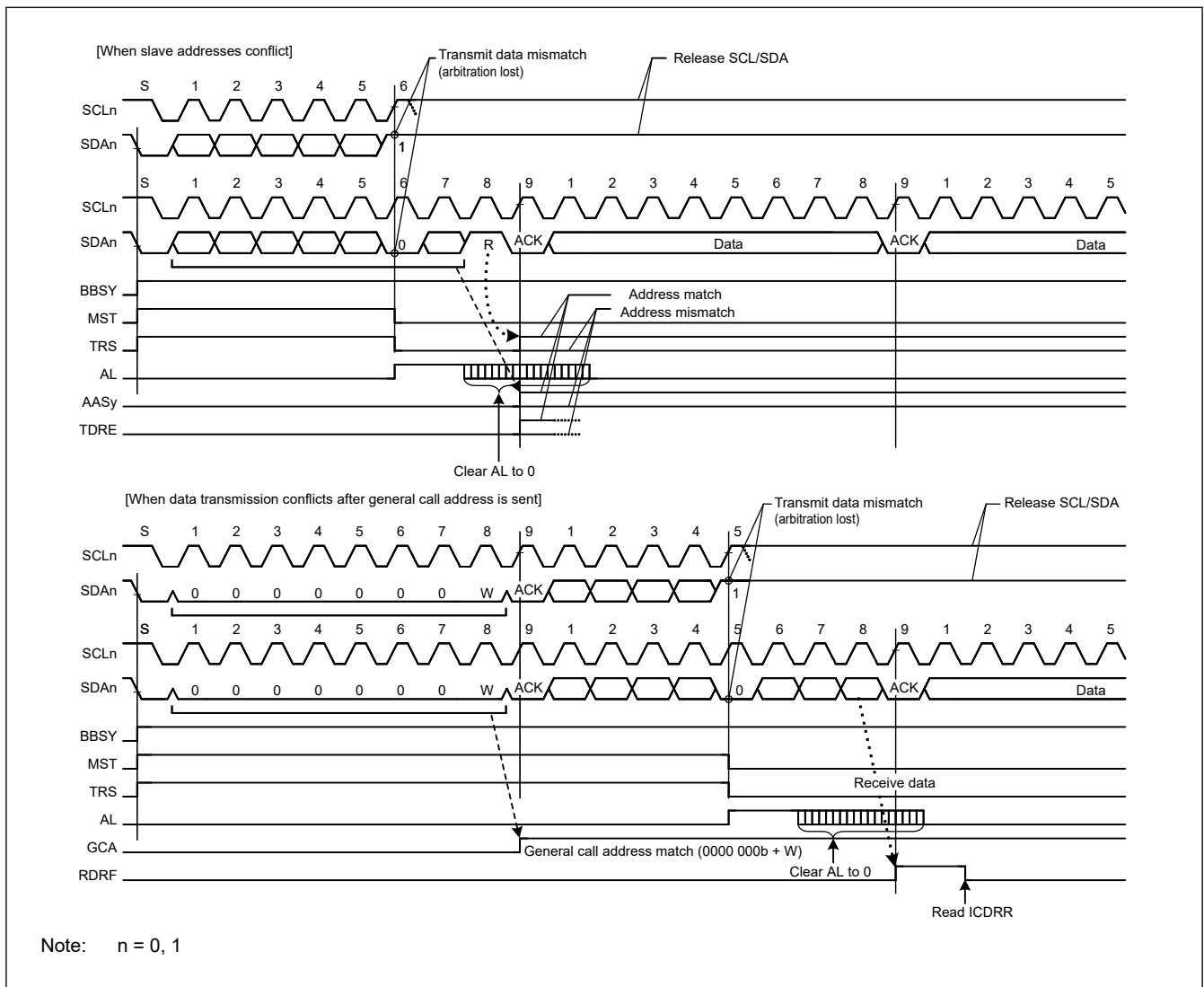
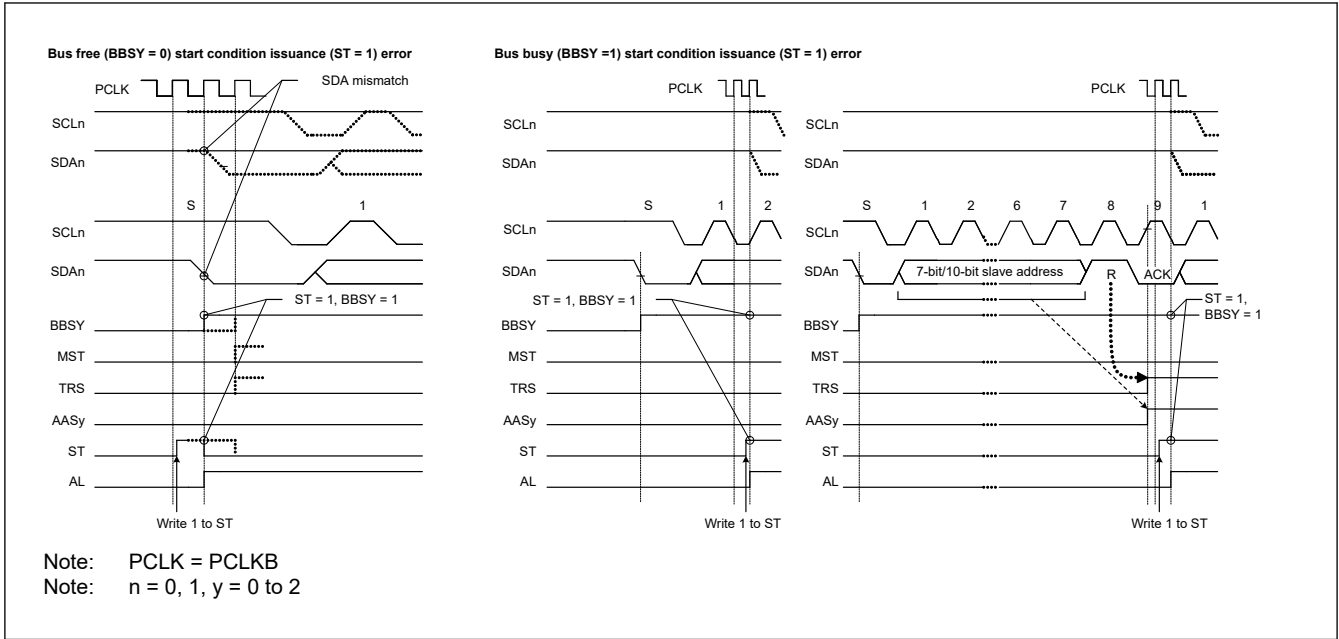


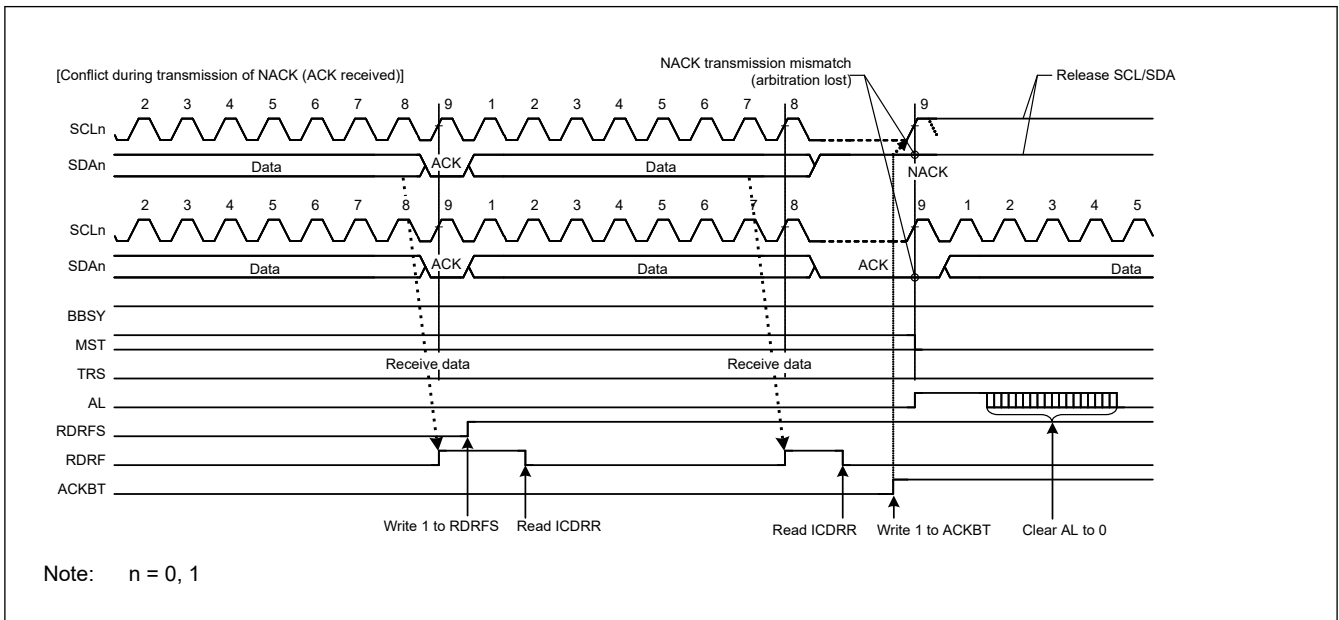
Figure 31.41 Examples of master arbitration-lost detection when MALE = 1



**Figure 31.42 Arbitration-lost when start condition is issued when MALE = 1**

### 31.10.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

This function causes arbitration to be lost if the internal SDA output level does not match the level on the SDA<sub>n</sub> line (high output as the internal SDA output, meaning the SDA<sub>n</sub> pin is in the high-impedance state) and the low level is detected on the SDA<sub>n</sub> line during transmission of NACK in receive mode. Arbitration is lost because of a conflict between NACK and ACK transmissions when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such a conflict occurs when multiple master devices send or receive the same information through a single slave device. [Figure 31.43](#) shows an example of arbitration-lost detection during transmission of NACK.



**Figure 31.43 Example of arbitration-lost detection during transmission of NACK when NALE = 1**

The following explains arbitration-lost detection using an example in which two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in either master A or master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Master A sends NACK when it has received 2 final bytes of



data from the slave device. Meanwhile, master B sends ACK because it has not received the required 4 bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect the ACK transmitted by master B and issues a stop condition. The stop condition issue conflicts with the SCL clock output of master B, which disrupts communication.

When the IIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost. If arbitration is lost during transmission of NACK, the IIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing, such as 0xFF transmission processing, which is required if the UDID (Unique Device Identifier) of the assigned address does not match in the Get UDID general processing after the Assign Address command.

The IIC detects arbitration-lost during transmission of NACK when the following condition is met with the NALE bit in ICFER set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDA<sub>n</sub> line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3).

### 31.10.3 Slave Arbitration-Lost Detection (SALE Bit)

This function causes arbitration to be lost if the transmit data (internal SDA output level) and the level on the SDA<sub>n</sub> line do not match (high output as the internal SDA output, meaning the SDA<sub>n</sub> pin is in the high-impedance state), and the low level is detected on the SDA<sub>n</sub> line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

When the IIC loses slave arbitration, the IIC is immediately released from the slave-matched state and enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminates subsequent redundant processing for the transmission of 0xFF.

The IIC detects slave arbitration-lost when the following condition is met with the SALE bit in ICFER set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDA<sub>n</sub> line in slave transmit mode (MST and TRS bits = 01b in ICCR2).

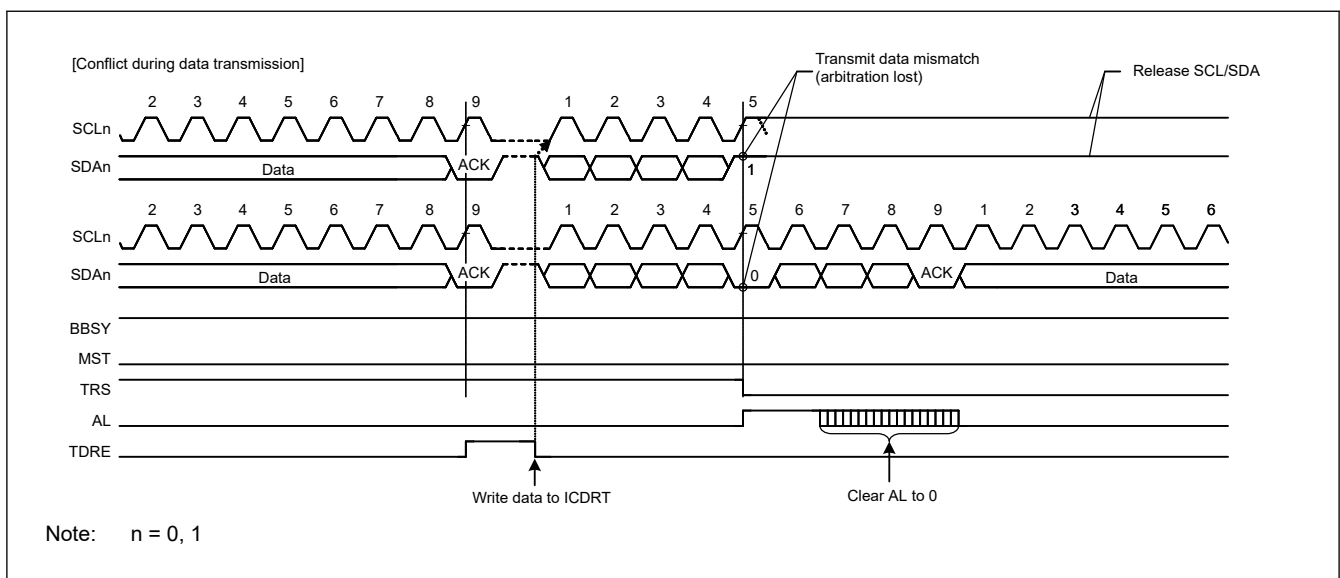


Figure 31.44 Example of slave arbitration-lost detection when SALE = 1

## 31.11 Start, Restart, and Stop Condition Issuing Function

### 31.11.1 Issuing a Start Condition

The IIC issues a start condition when the ST bit in ICCR2 is set to 1. When the ST bit is set to 1, a start condition request is made, and the IIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free state). When a start condition is issued normally, the IIC automatically shifts to the master transmit mode.

To issue a start condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure that the time set in ICBRH and the start condition hold time elapse.
3. Drive the SCLn line low (high level to low level).
4. Detect low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

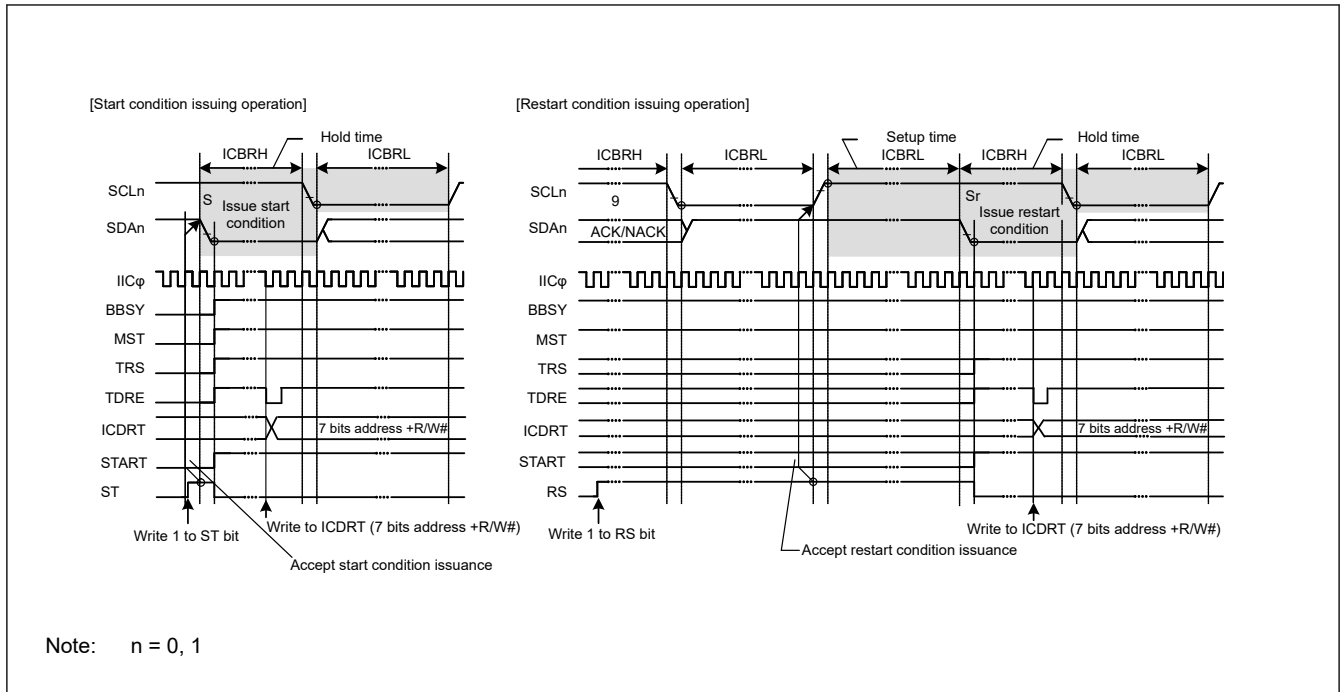
### 31.11.2 Issuing a Restart Condition

The IIC issues a restart condition when the RS bit in ICCR2 is set to 1. When the RS bit is set to 1, a restart condition request is made, and the IIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a restart condition:

1. Release the SDAn line.
2. Ensure the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRL and the restart condition setup time elapse.
5. Drive the SDAn line low (high level to low level).
6. Ensure the time set in ICBRH and the restart condition hold time elapse.
7. Drive the SCLn line low (high level to low level).
8. Detect a low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

**Note:** When issuing restart condition requests, write the slave address to ICDRT after confirming that ICCR2.RS = 0. Data written while ICCR2.RS = 1 is not forwarded because of the retransmission condition before the occurrence.



**Figure 31.45 Start and restart condition issue timing using the ST and RS bits**

Figure 31.46 shows the operation timing when a restart condition is issued after the master transmission.

[To issue a restart condition after the master transmission:]

1. Initialize the IIC using the procedure in [section 31.3.2. Initial Settings](#).
2. Read the BBSY flag in ICCR2 to check that the bus is open, then set the ST bit in ICCR2 to 1 (start condition issuance request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and the START flags in ICSR2 are automatically set to 1 and the ST bit is automatically set to 0. If the start condition is detected and the internal levels for the SDA output state and the levels on the SDA<sub>n</sub> line match while the ST bit is 1, the IIC recognizes that a start condition is successfully issued as requested by the ST bit has been successfully completed. The MST and TRS bits in ICCR2 are automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 when the TRS bit is set to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. After the data for transmission is written to ICDRT, the TDRE flag is automatically set to 0, data is transferred from ICDRT to ICDRS, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode according to the value of the transmitted R/W# bit. If the value of the R/W# bit is 0, the IIC continues in master transmit mode. If the ICSR2.NACKF flag is 1 at this time, indicating that no slave device recognized the address or that there was an error in communications, write 1 to ICCR2.SP bit to issue a stop condition. To transmit data with an address in the 10-bit format, start by writing 1111 0b, the 2 upper bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the 8 lower bits of the slave address to ICDRT.
4. After confirming that the TDRE flag in ICSR2 is 1, write data for transmission to the ICDRT register. The IIC automatically holds the SCL<sub>n</sub> line low until data for transmission is ready, a restart condition is issued or a stop condition is issued.
5. After all bytes of data for transmission are written to the ICDRT register, wait until the value of the TEND flag in ICSR2 returns to 1. Then after checking that the START flag in ICSR2 is 1, set the START flag in ICSR2 to 0.
6. Set the RS bit in ICCR2 to 1 (restart condition issue request). On receiving the request, the IIC issues a restart condition.
7. After checking that the START flag in ICSR2 is 1, write the value for transmission (the slave address and the R/W# bit) to ICDRT.

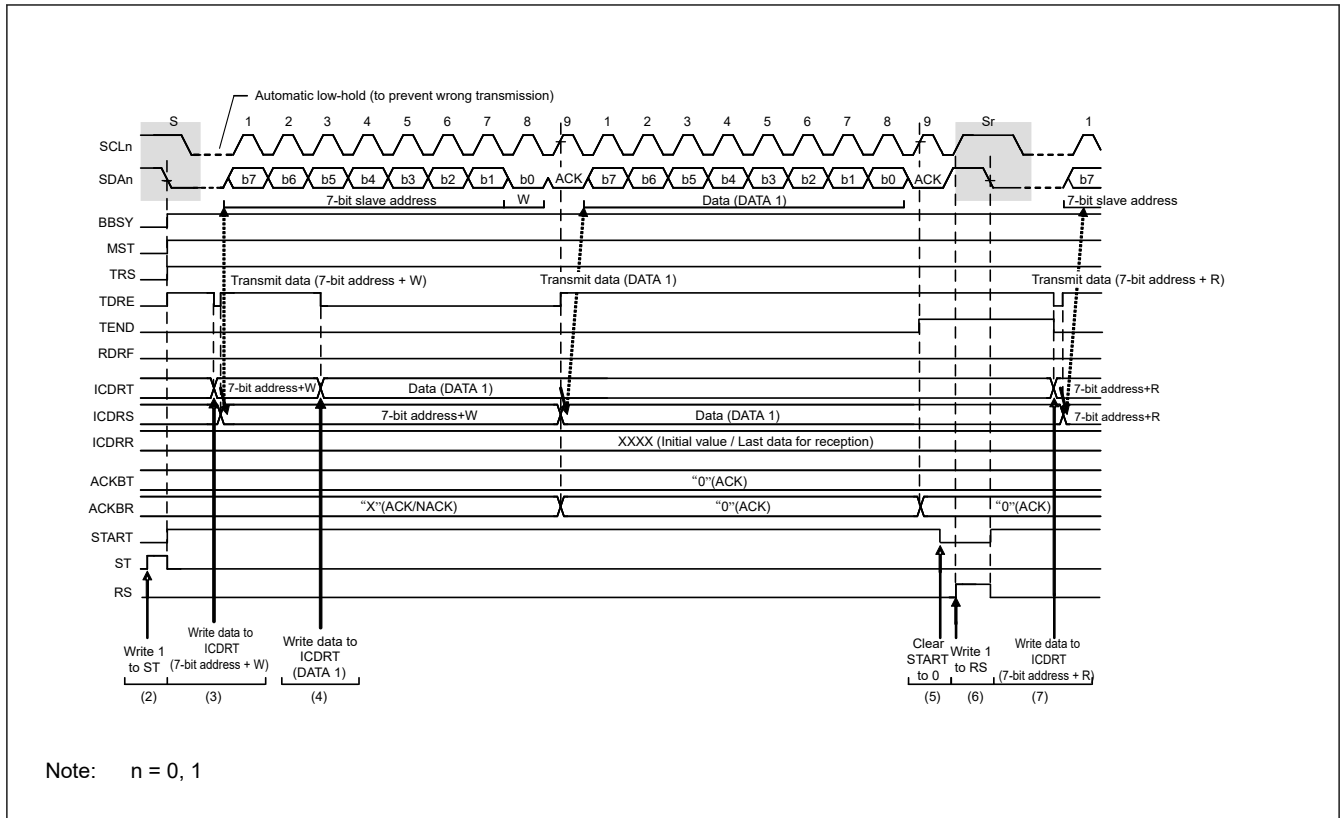


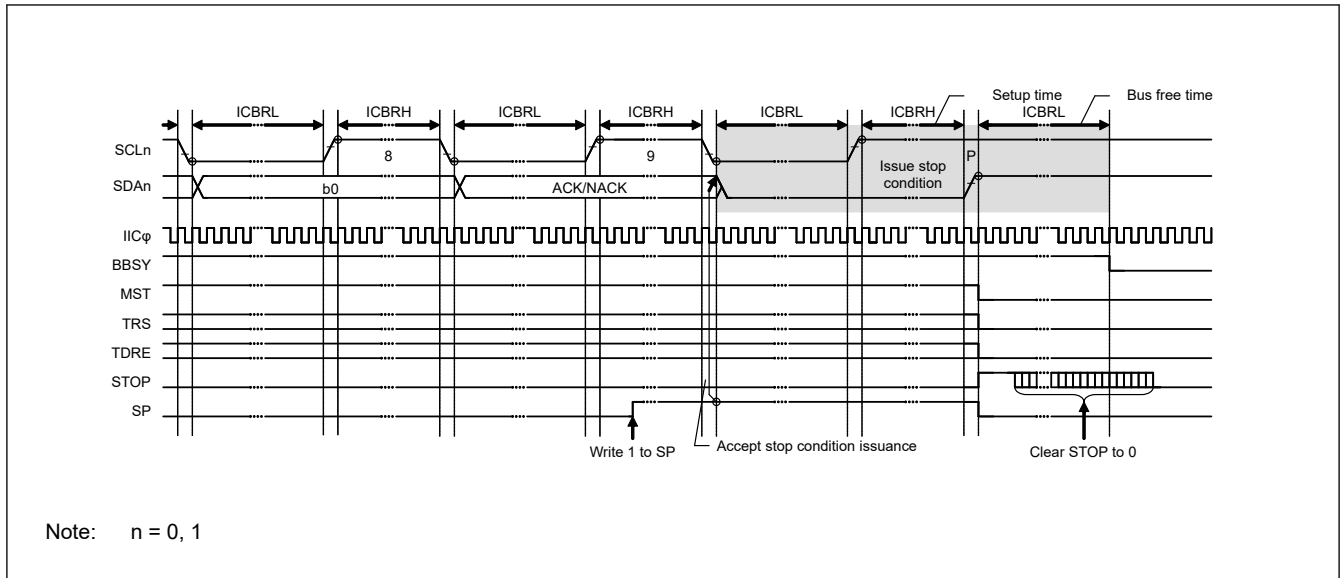
Figure 31.46 Restart condition issue timing after master transmission.

### 31.11.3 Issuing a Stop Condition

The IIC issues a stop condition when the SP bit in ICCR2 is set to 1. When the SP bit is set to 1, a stop condition request is made, and the IIC issues a stop condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a stop condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRH and the stop condition setup time elapse.
5. Release the SDAn line (low level to high level).
6. Ensure the time set in ICBRL and the bus free time elapse.
7. Clear the BBSY flag to 0 to release the bus mastership.



**Figure 31.47 Stop condition issue timing using the SP bit**

## 31.12 Bus Hanging

If the clock signals from the master and slave devices are out of synchronization because of noise or other factors, the I<sup>2</sup>C bus might hang with a fixed level on the SCLn line or SDAn line.

To manage bus hanging, the IIC has a timeout function to detect hanging by monitoring the SCLn line, and a function for outputting an extra SCL clock cycle to release the bus from:

- A timeout function to detect hanging by monitoring the SCLn line
- The IIC reset function
- An internal reset function.

By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the IIC or its communicating partner is placing the low level on the SCLn or SDAn line.

### 31.12.1 Timeout Function

The timeout function can detect when the SCLn line is stuck longer than the predetermined time. The IIC can detect an abnormal bus state by monitoring that the SCLn line is stuck low or high for a predetermined time.

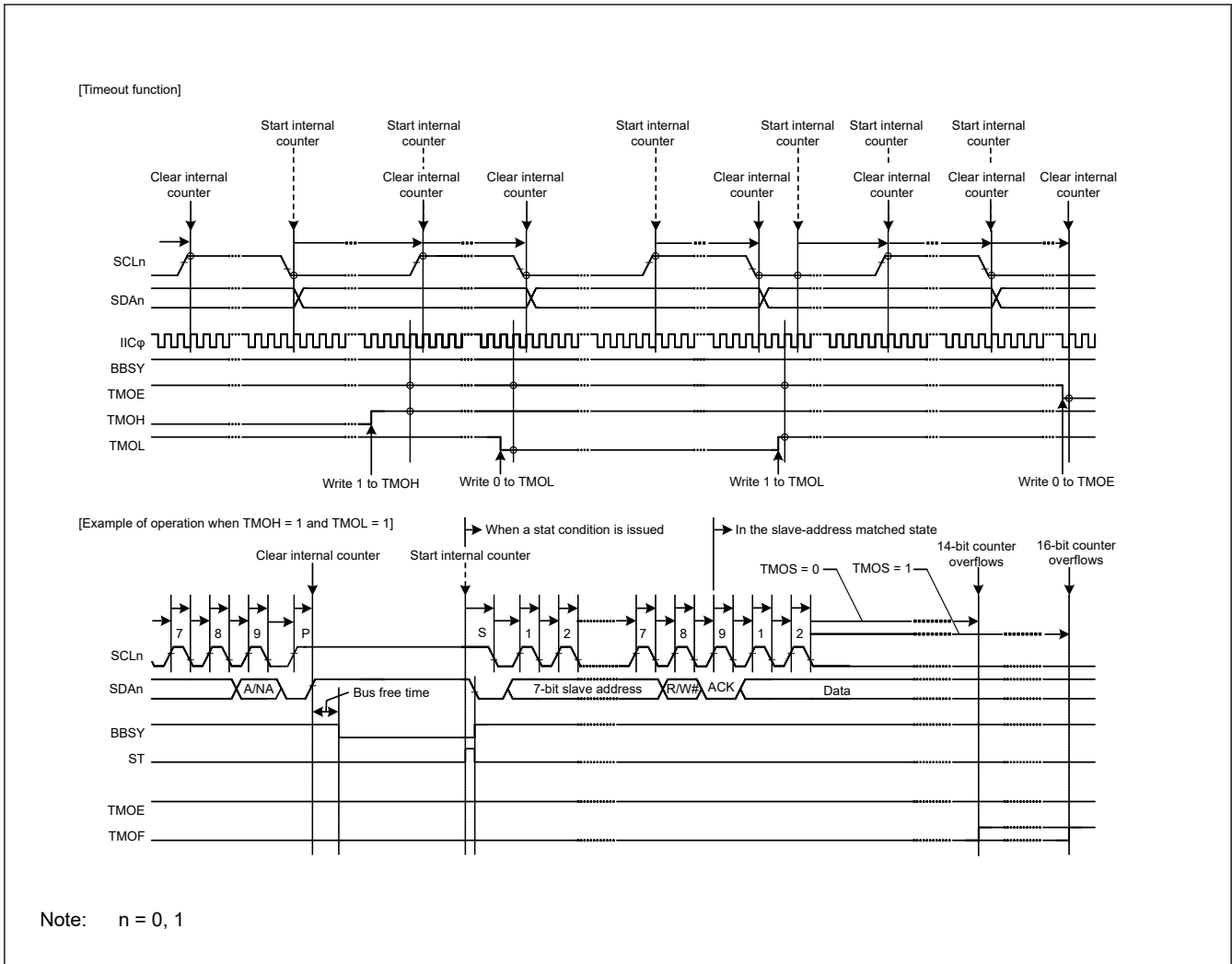
The timeout function monitors the SCLn line state and counts the low- or high-level period using the internal counter. The timeout function resets the internal counter each time the SCLn line changes (rises or falls), but continues to count unless the SCLn line changes. If the internal counter overflows because no SCLn line changes, the IIC can detect the timeout and report the bus hung state.

This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state when the SCLn line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1)
- The IIC slave address is detected (ICSR1 register is not 0x00) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0)
- The bus is open (ICCR2.BBSY flag is 0) while a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function uses the internal reference clock (IICφ) set in the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCLn line level (low, high, or both levels) during which this counter is activated can be selected in the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are set to 0, the internal counter is disabled.



**Figure 31.48** Timeout function using the TMOE, TMOS, TMOH, and TMOL bits

### 31.12.2 Extra SCL Clock Cycle Output Function

In master mode, this function outputs extra SCL clock cycles to release the SDAn line of the slave device from being held low because the master is out of synchronization with the slave device. This function is mainly used in master mode to release the SDAn line of the slave device from being fixed low by including extra cycles of SCL output from the IIC. It uses single cycles of the SCL clock for a bus error where the IIC cannot issue a stop condition because the slave device is holding the SDAn line at the low level. Do not use this function in normal situations. Using it when communications are proceeding correctly leads to malfunctions.

When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the frequency specified in the CKS[2:0] bits in ICMR1, and in the ICBRH and ICBRL registers, is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically set to 0. At this time, if ICCR2.BBSY = 1, the SCL pin goes low, and when ICCR2.BBSY = 0, the SCL pin goes high. After confirming that the CLO bit is 0 by software, write 1 to the CLO bit to output the additional clock continuously.

When the IIC module is in master mode and the slave device is holding the SDAn line low because synchronization with the slave device is lost because of effects like noise, the output of a stop condition is not possible. This function can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held low, and so recover the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming the release of the SDAn line by the slave device, complete communications by reissuing the stop condition.

[Output conditions for using the CLO bit in ICCR1]

- When the bus is open (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)

- When the communication device does not hold the SCLn line low.

Figure 31.49 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

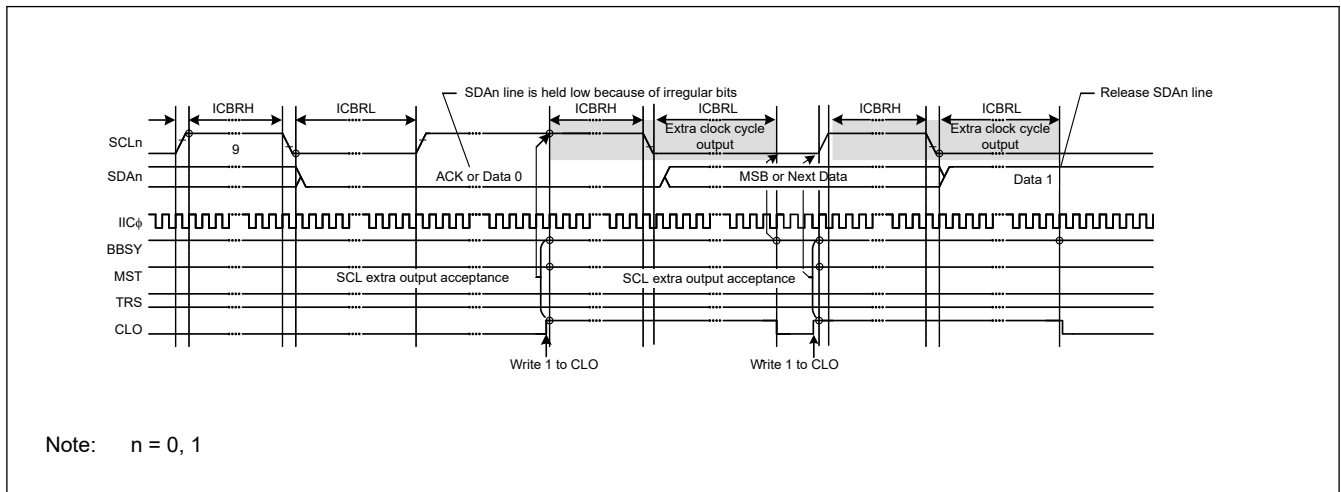


Figure 31.49 Extra SCL clock cycle output function using the CLO bit

### 31.12.3 IIC Reset and Internal Reset

The IIC module incorporates a function for resetting itself. It uses two types of resets:

- An IIC reset, which initializes all registers, including the BBSY flag in ICCR2.
- An internal reset, which releases the IIC from the slave-address matched state and initializes the internal counter while saving other settings.

After issuing a reset, always set the IICRST bit in ICCR1 to 0. Both types of resets are valid for release from bus-hung states, because both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Issuing a reset during slave operation might lead to a loss of synchronization between the master device clock and the slave device clock, so avoid this when possible. In addition, monitoring of the bus state, such as for the presence of a start condition, is not possible during an IIC reset (ICE and IICRST bits = 01b in ICCR1).

For a detailed description of the IIC and internal resets, see [section 31.15. State of Registers When Issuing Each Condition](#).

## 31.13 SMBus Operation

The IIC supports data communication conforming to the SMBus Specification (version 2.0). To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 to 100 kbps of the SMBus standard, set the CKS[2:0] bits in ICMR1, the ICBRH, and ICBRL registers. In addition, specify the values in the DLCS bit in ICMR2 and the SDDL[2:0] bits in ICMR2 to meet the data hold time specification of 300 ns or more. When the IIC is used only as a slave device, the transfer rate setting is not required, but ICBRL must be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (SARL0, SARL1, and SARL2), and set the associated FS bit (7- or 10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

### 31.13.1 SMBus Timeout Measurement

#### (1) Measuring slave device timeout

The following period (timeout interval:  $T_{LOW:SEXT}$ ) must be measured for slave devices in SMBus communication:

- From start condition to stop condition.

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the GPT using the IIC start condition detection interrupt (STIn) and stop condition detection interrupt (SPIn). The measured timeout period must be within the total clock low-level period [slave device]  $T_{LOW:SEXT}$ : 25 ms (maximum) of the SMBus standard.

If the time measured with the GPT exceeds the clock low-level detection timeout  $T_{TIMEOUT}$ : 25 ms (minimum) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the IIC. When an internal reset is issued, the IIC stops driving the bus for the SCLn and SDAn pins, making them output high-impedance, which releases the bus.

(2) Measuring master device timeout

The following periods (timeout interval:  $T_{LOW:MEXT}$ ) must be measured for master devices in SMBus communication:

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition.

To measure timeout for master devices, measure these periods with the GPT using the IIC start condition detection interrupt (STIn), stop condition detection interrupt (SPIn), transmit end interrupt (IICn\_TEI), or receive data full interrupt (IICn\_RXI). The measured timeout period must be within the total clock low-level extended period (master device)  $T_{LOW:MEXT}$ : 10 ms (maximum) of the SMBus standard, and the total of all  $T_{LOW:MEXT}$  values from start condition to stop condition must be within  $T_{LOW:SEXT}$ : 25 ms (maximum).

For the ACK receive timing (rising edge of the 9th SCL clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). Perform byte-wise transmit operations in master transmit mode, and hold the RDRFS bit in ICMR3 at 0 until the byte immediately before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

If the period measured with the GPT exceeds the total clock low-level extended period (master device)  $T_{LOW:MEXT}$ : 10 ms (maximum) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout  $T_{TIMEOUT}$ : 25 ms (minimum) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (stop writing data to ICDRT).

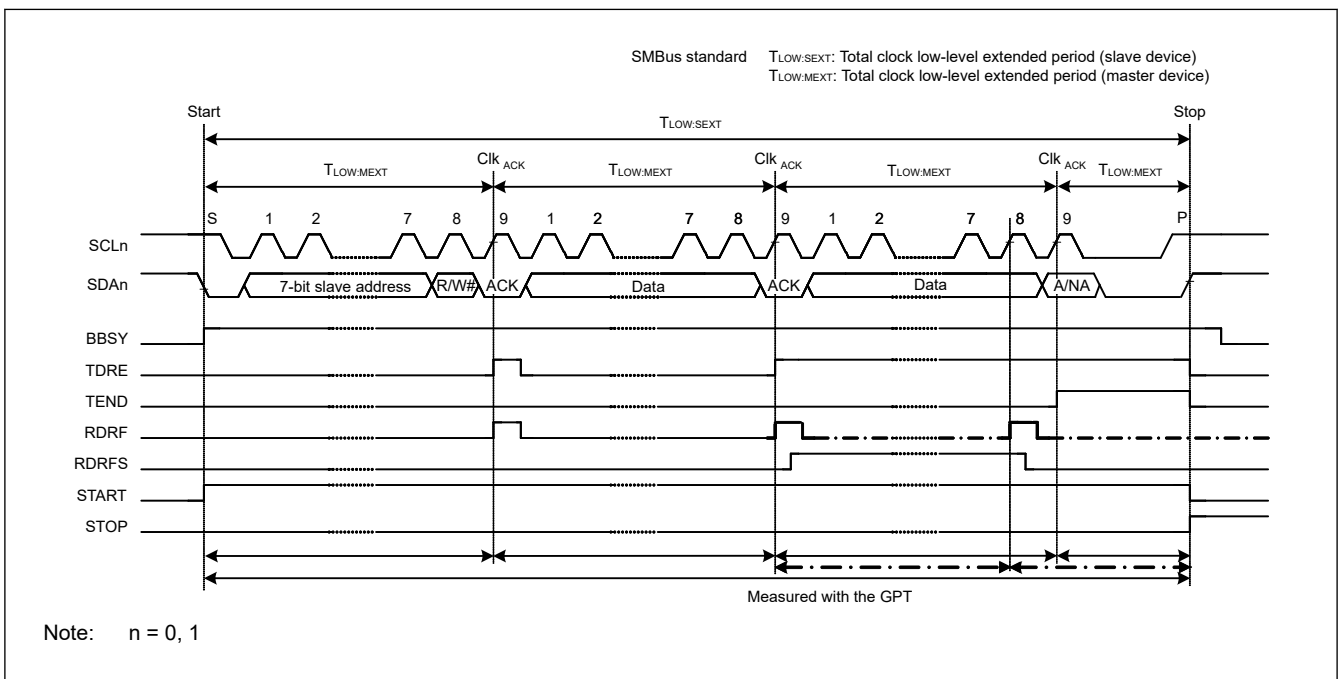


Figure 31.50 SMBus timeout measurement



### 31.13.2 Packet Error Code (PEC)

The MCU provides a CRC calculator that enables transmission of a Packet Error Code (PEC) or allows checking of the received data in SMBus data communication for the IIC. For the CRC-generating polynomials of the CRC calculator, see [section 38, Cyclic Redundancy Check \(CRC\)](#).

In master transmit mode, the PEC data can be generated by writing all transmit data to the CRC Data Input Register (CRCDIR) in the CRC calculator.

In master receive mode, the PEC data can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC Data Output Register (CRCDOR) with the received PEC data.

To send ACK or NACK based on the match or mismatch result when the final byte is received as a result of the PEC code check, set the RDRFS bit in ICMR3 to 1 before the rising edge of the 8th SCL clock cycle during reception of the final byte, and hold the SCLn line low on the falling edge of the 8th clock cycle.

### 31.13.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communicating over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address, or to request its own slave address from the SMBus host.

For a product using the MCU to operate as an SMBus host or ARP master, the host address (0001 000b) sent from the slave device must be detected as a slave address, and so the IIC provides a function for detecting the host address. To detect the host address as a slave address, set the SMBS bit in ICMR3 and the HOAE bit in ICSE1 to 1. Operation after the host address is detected is the same as normal slave operation.

## 31.14 Interrupt Sources

The IIC issues five types of interrupt requests:

- Transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start condition detection, and stop condition detection)
- Receive data full
- Transmit data empty
- Transmit end
- Address match during wakeup function

[Table 31.10](#) lists details about the interrupt requests. The receive data full and transmit data empty interrupts can activate data transfer by the DTC or DMAC.

**Table 31.10 Interrupt sources**

Symbol	Interrupt source	Interrupt flag	DTC or DMAC activation	Interrupt condition
IICn_EEI* <sup>5</sup>	Transfer error or event occurrence	AL	Not possible	AL = 1, ALIE = 1
		NACKF		NACKF = 1, NAKIE = 1
		TMOF		TMOF = 1, TMOIE = 1
		START		START = 1, STIE = 1
		STOP		STOP = 1, SPIE = 1
IICn_RXI* <sup>2</sup> * <sup>5</sup>	Receive data full	RDRF	Possible	RDRF = 1, RIE = 1
IICn_TXI* <sup>1</sup> * <sup>5</sup>	Transmit data empty	RDRF	Possible	TDRE = 1, TIE = 1
IICn_TEI* <sup>3</sup> * <sup>5</sup>	Transmit end	TEND	Not possible	TEND = 1, TEIE = 1
IIC0_WUI* <sup>4</sup>	Slave address match during wakeup function	WUF	Not possible	Slave address match Slave receive complete RWAK operation ASY0 = 1 WUIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and the actual writing to the module. When an interrupt flag is cleared or masked, read the relevant flag again to check whether clearing or masking is complete, then return from interrupt handling. Not doing so creates the possibility of repeated processing of the same interrupt.

- Note 1. Because IICn\_TXI is edge-detected, it does not require clearing. Additionally, the TDRE flag in ICSR2 (condition for IICn\_TXI) is automatically set to 0 when transmit data is written to the ICDRT register or a stop condition is detected (STOP flag = 1 in ICSR2).
- Note 2. Because IICn\_RXI is edge-detected, it does not require clearing. Additionally, the RDRF flag in ICSR2 (condition for IICn\_RXI) is automatically set to 0 when data is read from ICDRR.
- Note 3. When using the IICn\_TEI interrupt, clear the TEND flag in ICSR2 in the IICn\_TEI interrupt handling. The TEND flag in ICSR2 automatically is set to 0 when transmit data is written to the ICDRT register or a stop condition is detected (STOP flag = 1 in ICSR2).
- Note 4. Only channel 0 has a wakeup function, so IIC0\_WUI is for channel 0 only.
- Note 5. Channel number (n = 0 to 1).

Clear or mask each flag during interrupt handling.

### 31.14.1 Buffer Operation for IICn\_TXI and IICn\_RXI Interrupts

If the conditions for generating an IICn\_TXI or IICn\_RXI interrupt are satisfied while the associated IR flag is 1, the interrupt request is not output for the ICU but is saved internally. One request per source can be saved internally.

An interrupt request that is saved in the ICU is output when the ICU.IELSRn.IR flag becomes 0. Internally saved interrupt requests are automatically cleared under normal conditions. They can also be cleared by writing 0 to the interrupt enable bit within the associated peripheral module.

### 31.15 State of Registers When Issuing Each Condition

The IIC has two dedicated resets, IIC reset and Internal reset. [Table 31.11](#) lists the registers states when issuing each condition.

**Table 31.11 Register states when issuing each condition (1 of 2)**

Registers		Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection	
ICCR1	ICE, IICRST	In reset	Saved	Saved	Saved	Saved	
	SCLO, SDAO		In reset	In reset			
	Others			Saved			
ICCR2	BBSY	In reset	In reset	Saved	Set	In reset	
	ST, RS			In reset	In reset		Set or saved
	SP						
	TRS						
	MST						
ICMR1	BC[2:0]	In reset	In reset	In reset	In reset	Saved	
	Others			Saved	Saved		
ICMR2		In reset	In reset	Saved	Saved	Saved	
ICMR3	ACKBT	In reset	In reset	Saved	Saved	In reset	
	Others					Saved	
ICFER		In reset	In reset	Saved	Saved	Saved	
ICSER		In reset	In reset	Saved	Saved	Saved	
ICIER		In reset	In reset	Saved	Saved	Saved	
ICSR1		In reset	In reset	In reset	Saved	In reset	
ICSR2	TEND	In reset	In reset	In reset	Saved	In reset	
	TDRE				Set or saved		
	START				Set		
	STOP				Saved		Set
	Others				Saved		Saved
ICWUR		In reset	In reset	Saved	Saved	Saved	

**Table 31.11 Register states when issuing each condition (2 of 2)**

Registers	Reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection	
SARL0, SARL1, SARL2 SARU0, SARU1, SARU2	In reset	In reset	Saved	Saved	Saved	
ICBRH, ICBRL	In reset	In reset	Saved	Saved	Saved	
ICDRT	In reset	In reset	Saved	Saved	Saved	
ICDRR	In reset	In reset	Saved	Saved	Saved	
ICDRS	In reset	In reset	In reset	Saved	Saved	
Timeout function	In reset	In reset	In reset	Operating	Operating	
Bus free time measurement	In reset	In reset	Operating	Operating	Operating	
ICWUR2	WUSEN	In reset	In reset	Saved	Saved	Saved
	Others					Saved or set or reset

### 31.16 Event Link Output

The IIC0 module handles the event output for the Event Link Controller (ELC) for the following sources:

#### (1) Transfer error event

When a transfer error event occurs, the associated event signal can be output to another module by the ELC.

#### (2) Receive data full

When a receive data register becomes full, the associated event signal can be output to another module by the ELC.

#### (3) Transmit data empty

When a transmit data register becomes empty, the associated event signal can be output to another module by the ELC.

#### (4) Transmit end

On completion of the transfer, the associated event signal can be output to another module by the ELC.

### 31.16.1 Interrupt Handling and Event Linking

Each of the IIC interrupt types (see [Table 31.10](#)) has an enable bit to control enabling and disabling of the associated interrupt signal. An interrupt request signal is output to the CPU when an interrupt source condition is satisfied while the associated enable bit is set.

The associated event link output signals are sent to other modules as event signals by the ELC when the interrupt source conditions are satisfied, regardless of the interrupt enable bit settings. For details on interrupt sources, see [Table 31.10](#).

### 31.17 Usage Notes

#### 31.17.1 Settings for the Module-Stop Function

The Module Stop Control Register B (MSTPCRB) can enable or disable IIC operation. The IIC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

#### 31.17.2 Notes on Starting Transfer

If the IR flag associated with the IIC interrupt is 1 when transfer is started (ICCR1.ICE bit = 1), follow the procedure in this section to clear the interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally saved after transfer starts, and this can lead to unexpected behavior of the IR flag.

To clear interrupts before starting transfer operation:

1. Confirm that the ICCR1.ICE bit is 0.

2. Set the relevant interrupt enable bits, such as ICIER.TIE to 0.
3. Read the relevant interrupt enable bits, such as ICIER.TIE, and confirm that the value is 0.
4. Set the IR flag to 0.

## 32. CAN with Flexible Data-rate (CANFD)

### 32.1 Overview

The CAN with Flexible Data-rate (CANFD) supports the following functions:

- CAN with Flexible Data-rate.\*<sup>1</sup>

Note 1. This feature is not available in the classical CAN function.

The CANFD module has a flexible message buffer and FIFO structure that meet the requirements of various applications. It also provides test modes to achieve high testability of the module that can be useful for power-on testing.

This specification describes of the CANFD module.

The CANFD mode is only available in certain products that support it.

#### 32.1.1 CANFD Module

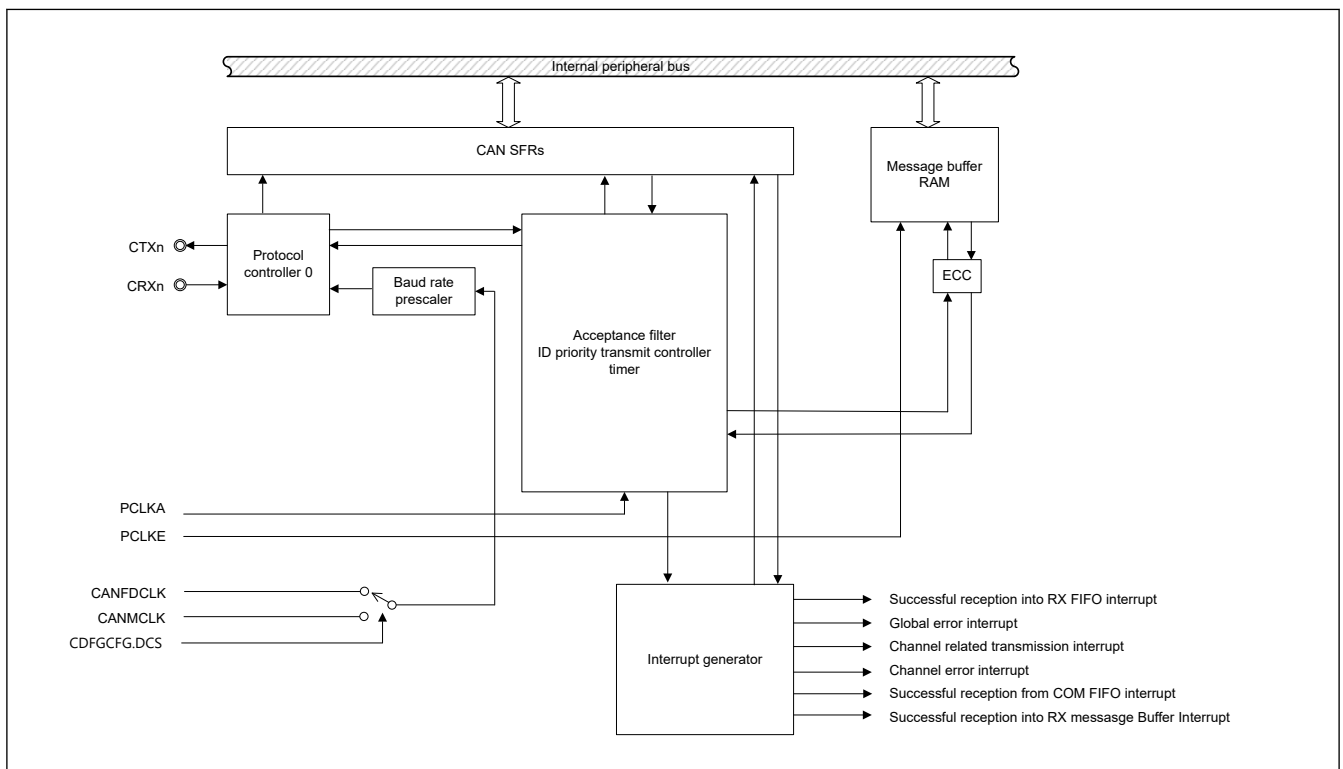
**Table 32.1 CANFD module specifications (1 of 2)**

Parameter		Specifications
Communication		CAN functionality conforms to CANFD ISO 11898-1 (2015)
Protocol engine version		RS-CANFD_PE V3.0
Data transfer rate	CANFD* <sup>1</sup>	Up to 1 Mbps for arbitration phase and up to 8 Mbps for data phase, individually for CAN channel
	Classical CAN	Up to 1 Mbps
Operation frequency of Peripheral module clock		120 MHz (PCLKA) RAM clock: 240 MHz (PCLKE)
Data Link Layer (DLL) clock		8 MHz to 80MHz The following clocks can be selected. <ul style="list-style-type: none"> <li>• CANMCLK : External Oscillator Clock</li> <li>• CANFDCLK : CANFD Core Clock</li> </ul>
Input/Output pins		CRXn/CTXn (n = 0, 1)
CAN channels		2 channels
Selectable ID type		11-bit Standard ID
		11-bit Standard ID + 18-bit Extended ID
Selectable frame type		Data frame (RTR = 0) (CAN and CANFD frames)
		Remote frame (RTR = 1) (only CAN frames)
Variable data byte count for data frames		DLC range: 0 to F
Message buffer		Up to 16 reception message buffers
		4 transmit message buffers per channel
		1 transmission queue per channel Automatic message transfer into transmission queues supported
FIFO number		2 reception FIFO buffers 1 COMMON FIFO individually configurable as: <ul style="list-style-type: none"> <li>• Reception FIFO</li> <li>• Transmission FIFO</li> </ul>
Automatic delay interval timer for transmission		The delay timer can be applied to: <ul style="list-style-type: none"> <li>• Transmission FIFO</li> </ul>

**Table 32.1 CANFD module specifications (2 of 2)**

Parameter	Specifications
Enhanced reception filtering	Support of 11 bits and 29 bits CAN identifier
	Programmable 29 bits CAN identifier acceptance filter mask for each entry
	Programmable routing capability for each FIFO and reception message buffers (up to 2 routing destinations)
	RTR and IDE masking
	Data Length Code (DLC) filter
	Message buffer payload overload protection
	Payload filter
	Updating Acceptance Filter List (AFL) entry during communication
General software support	Automatic label information added to receive message (for upper software layer support)
Timer	TX and RX Time Stamp function
Power down function	Module start stop function for each CAN node (Channel and Global Sleep mode)
RAM	RAM ECC protected (2 bits error detection, 1-bit error correction)
Module-stop function	Module-stop state can be set for each channels to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set for each channels.

Note 1. The CANFD mode is available only for CANFD supported product.



**Figure 32.1 Overview of the CANFD module**

- CRXn/CTXn:  
Input/Output pins of the CANFD module
- Protocol controller:  
Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, error handling

- Message buffer RAM (MBRAM):  
This RAM is used to store messages after reception or for transmission using a normal message buffer or a FIFO. Each message entry has an individual ID, data length code, data field, message pointer for upper layer application usage and a time stamp.  
This RAM is used to store the message acceptance filtering entries. Each acceptance filter entry has an individual ID, data length code, data field, message pointer for upper layer application usage and message direction pointer.
- Acceptance filter list RAM (AFLRAM):  
Performs filtering of received messages. The entries in the Acceptance filter list RAM are used for the filtering process.
- Two timers:
  - Reception Timestamp function
  - Transmission separation time for FIFO buffers
- Interrupt generator:  
Generates several types of global and channel interrupts
- CAN Special Function Registers (SFRs):  
Registers associated with CAN. See [section 32.2. Register Descriptions](#).

### 32.1.2 Clock restriction

For the CAN communication the following restriction for the clocks should be satisfied:

- $PCLKE / 2 = PCLKA \geq CANFDCLK$
- $PCLKE / 2 = PCLKA \geq CANMCLK$

To avoid missing events the CAN engine clock (CANFDCLK or CANMCLK) frequency must be less than the PCLKA clock frequency.

To avoid loss of CAN message, the PCLKA should be set to a clock with a frequency depend on the CAN communication Baud Rate. The constraint of a baud rate and a PCLKA clock is shown in [Table 32.2](#).

**Table 32.2 Clock restriction**

	Baud rate	PCLKA
CANFD	1 Mbps Nominal 8 Mbps Data	PCLKA $\geq$ 40 MHz
	500 Kbps Nominall 5 Mbps Data	PCLKA $\geq$ 32 MHz
Classical CAN	1 Mbps Data	PCLKA $\geq$ 32 MHz

The frequency of CANFDCLK and CANMCLK depend on the required baud rate. For information how to configure the baud rate, refer to [section 32.4.1.3. Baud Rate](#).

## 32.2 Register Descriptions

### 32.2.1 Register Table

The reset value shown for the RAM area, consisting of CFDGAFLLIDr, CFDGAFLLMr, CFDGAFLLP0r, CFDGAFLLP1r, CFDRMBBCPb, CFDRFMBBCPb, CFDCFMBBCP0, CFDTMBBCPb, CFDTHLACC0, CFDTHLACC1 and CFDRPGACCK is valid after initialization of a hardware reset. See [section 32.4.2. CAN Module Configuration after Hardware Reset](#) for details of the initialization process.

If a write access with a size of 8 or 16 bits is performed for the RAM area, then the CANFD module does a read-modify write-access to the RAM location, because the RAM requires a 32-bit access through the ECC module.

For single bit error, the correct data is written back. For multiple bit errors, unknown data is written back.

Do not access the space where the register is not assigned.

The read data from the space where the register is not assigned is unknown.

### 32.2.2 CFDC0NCFG : Nominal Bitrate Configuration Register

Base address: CANFD<sub>n</sub> = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFD<sub>n</sub>\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0000

Bit position:	31	25 24	17 16	10 9	0
Bit field:	NTSEG2[6:0]		NTSEG1[7:0]		NBRP[9:0]

Value after reset: 0

Bit	Symbol	Function	R/W
9:0	NBRP[9:0]	Channel Nominal Baud Rate Prescaler Nominal baud rate prescaler division ratio	R/W
16:10	NSJW[6:0]	Resynchronization Jump Width 0x00: 1 Tq 0x01: 2 Tq ⋮ 0x7E: 127 Tq 0x7F: 128 Tq	R/W
24:17	NTSEG1[7:0]	Timing Segment 1 0x00: Reserved 0x01: 2 Tq 0x02: 3 Tq 0x03: 4 Tq ⋮ 0xFE: 255 Tq 0xFF: 256 Tq	R/W
31:25	NTSEG2[6:0]	Timing Segment 2 0x00: Reserved 0x01: 2 Tq ⋮ 0x7E: 127 Tq 0x7F: 128 Tq	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Tq means time quantum.

This register configures the transmission/reception nominal baud rate parameters of the channels.

#### NBRP[9:0] bits (Channel Nominal Baud Rate Prescaler)

The NBRP[9:0] bits are used to define the peripheral bus clock periods contained in a time quantum.

Do not write to these bits in CH\_OPERATION or CH\_SLEEP mode.

Only write to these bits when the CANFD channel is in CH\_RESET or CH\_HALT mode.

#### NSJW[6:0] bits (Resynchronization Jump Width)

The NSJW[6:0] bits set the synchronization jump width. A value from 1 to 128 time quanta can be set.

Do not write to these bits in CH\_OPERATION or CH\_SLEEP mode.

Only write to these bits when the CANFD channel is in CH\_RESET or CH\_HALT mode.

#### NTSEG1[7:0] bits (Timing Segment 1)

The NTSEG1[7:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. These bits contain the propagation segment.

Do not write to these bits in CH\_OPERATION or CH\_SLEEP mode.

Only write to these bits when the CANFD channel is in CH\_RESET or CH\_HALT mode.

Additionally, configure a Tq value only between 2 and 256, inclusive. See [section 32.4.1.2. CAN Bit Timing](#) for more details.



**NTSEG2[6:0] bits (Timing Segment 2)**

The NTSEG2[6:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error.

Do not write to these bits in CH\_OPERATION or CH\_SLEEP mode.

Only write to these bits when the CANFD channel is in CH\_RESET or CH\_HALT mode.

Additionally, configure a Tq value only between 2 and 128, inclusive.

**32.2.3 CFDC0CTR : Control Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ROM	BFT	—	—	—	CTMS[1:0]	CTME	ERRD	BOM[1:0]	—	TDCV FIE	SOCO IE	EOCO IE	TAIE		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLPR	CHMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
1:0	CHMDC[1:0]	Channel Mode Control 0 0: Channel operation mode request 0 1: Channel reset request 1 0: Channel halt request 1 1: Keep current value	R/W
2	CSLPR	Channel Sleep Request 0: Channel sleep request disabled 1: Channel sleep request enabled	R/W
3	RTBO	Return from Bus-Off 0: Channel is not forced to return from bus-off 1: Channel is forced to return from bus-off	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt disabled 1: Bus error interrupt enabled	R/W
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt disabled 1: Error warning interrupt enabled	R/W
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt disabled 1: Error passive interrupt enabled	R/W
11	BOEIE	Bus-Off Entry Interrupt Enable 0: Bus-off entry interrupt disabled 1: Bus-off entry interrupt enabled	R/W
12	BORIE	Bus-Off Recovery Interrupt Enable 0: Bus-off recovery interrupt disabled 1: Bus-off recovery interrupt enabled	R/W
13	OLIE	Overload Interrupt Enable 0: Overload interrupt disabled 1: Overload interrupt enabled	R/W
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt disabled 1: Bus lock interrupt enabled	R/W

Bit	Symbol	Function	R/W
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt disabled 1: Arbitration lost interrupt enabled	R/W
16	TAIE	Transmission Abort Interrupt Enable 0: TX abort interrupt disabled 1: TX abort interrupt enabled	R/W
17	EOCOIE	Error Occurrence Counter Overflow Interrupt Enable 0: Error occurrence counter overflow interrupt disabled 1: Error occurrence counter overflow interrupt enabled	R/W
18	SOCOIE	Successful Occurrence Counter Overflow Interrupt Enable 0: Successful occurrence counter overflow interrupt disabled 1: Successful occurrence counter overflow interrupt enabled	R/W
19	TDCVFIE <sup>*1</sup>	Transceiver Delay Compensation Violation Interrupt Enable 0: Transceiver delay compensation violation interrupt disabled 1: Transceiver delay compensation violation interrupt enabled	R/W
20	—	This bit is read as 0. The write value should be 0.	R/W
22:21	BOM[1:0]	Channel Bus-Off Mode 0 0: Normal mode (comply with ISO 11898-1) 0 1: Entry to Halt mode automatically at bus-off start 1 0: Entry to Halt mode automatically at bus-off end 1 1: Entry to Halt mode (during bus-off recovery period) by software	R/W
23	ERRD	Channel Error Display 0: Only the first set of error codes displayed 1: Accumulated error codes displayed	R/W
24	CTME	Channel Test Mode Enable 0: Channel test mode disabled 1: Channel test mode enabled	R/W
26:25	CTMS[1:0]	Channel Test Mode Select 0 0: Basic test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (External loopback mode) 1 1: Self-test mode 1 (Internal loopback mode)	R/W
29:27	—	These bits are read as 0. The write value should be 0.	R/W
30	BFT	Bit Flip Test 0: First data bit of reception stream not inverted 1: First data bit of reception stream inverted	R/W
31	ROM <sup>*1</sup>	Restricted Operation Mode 0: Restricted operation mode disabled 1: Restricted operation mode enabled	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. These bits are not available in the classical CAN function.

Channel Control register controls the modes of the related channel. It is used to enable generation of interrupts if errors are detected on the CAN bus connected to this channel. It is also used to configure the channel in test mode.

### CHMDC[1:0] bits (Channel Mode Control)

The CHMDC[1:0] bits can be used to configure modes of the CAN channel.

CAN mode transitions are described in more details in [section 32.3.3. Channel Modes](#).

Setting CHMDC[1:0] bits to 11b has no effect. When the CANFD module is in GL\_HALT mode, these bits can only be set to 10b or 01b. These bits cannot be set in CH\_SLEEP mode.

These bits can change automatically when transitioning to Halt mode by the CFDC0CTR.BOM settings.

If CPU write access to CFDC0CTR.CHMDC occurs at the same time when the CAN channel enters Halt mode (at the start of bus-off when CFDC0CTR.BOM = 01b, or at the end of bus-off when CFDC0CTR.BOM = 10b), then the CPU write access has the highest priority.

The CAN channel changes the value of CFDC0CTR.CHMDC within the Channel Control Registers for the specified cases only if the CFDC0CTR.CHMDC value is 00b (Operation mode).

#### **CSLPR bit (Channel Sleep Request)**

When the CSLPR bit is 1, a Sleep mode request is generated for the corresponding CAN channel

When this bit is 0, a request to exit Sleep mode is generated for the related CANFD channel.

Only write to this bit when the related CANFD channel is in CH\_RESET or CH\_SLEEP mode.

#### **RTBO bit (Return from Bus-Off)**

When the protocol controller of the CAN channel enters bus-off state, you can force a recovery from bus-off state by setting the RTBO bit in the Channel Control Register to 1.

The error state changes from bus-off state to integrating with a maximum delay of 1 CAN bit time.

When the RTBO bit is set to 1, the REC and TEC registers are initialized and the Bus-Off Status bit (Channel Bus-off Status, CFDC0STS.BOSTS) is set to 0.

Registers other than the REC and TEC registers are not initialized by this command. Even if CFDC0CTR.BORIE is set, a bus-off recovery interrupt is not generated by this recovery from the bus-off state.

The RTBO bit cannot be set in CH\_SLEEP mode. Setting this bit in any state other than bus-off state has no effect and the bit is cleared immediately. The read value is always 0.

Return from the Bus-Off command should be used only when CFDC0CTR.BOM is set to 00b.

Only write to this bit when the related CANFD channel is in CH\_OPERATION mode. This bit is automatically cleared when set by software.

#### **BEIE bit (Bus Error Interrupt Enable)**

When the BEIE and the CFDC0ERFL.BEF bits are both 1, an error interrupt request is generated.

This bit cannot be set in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_RESET mode.

#### **EWIE bit (Error Warning Interrupt Enable)**

When the EWIE and the CFDC0ERFL.EWF bits are both 1, an error interrupt request is generated.

The EWIE bit cannot be set in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_RESET mode.

#### **EPIE bit (Error Passive Interrupt Enable)**

An error interrupt request is generated when the EPIE bit and the CFDC0ERFL.EPF are both 1.

The EPIE bit cannot be set in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_RESET mode.

#### **BOEIE bit (Bus-Off Entry Interrupt Enable)**

When the BOEIE and the CFDC0ERFL.BOEF bits are both 1, an error interrupt request is generated.

The BOEIE bit cannot be set in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_RESET mode.

#### **BORIE bit (Bus-Off Recovery Interrupt Enable)**

When the BORIE and the CFDC0ERFL.BORF bits are both 1, an error interrupt request is generated.

The BORIE bit cannot be set in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_RESET mode.

#### **OLIE bit (Overload Interrupt Enable)**

When the OLIE and the CFDC0ERFL.OVLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_RESET mode.

**BLIE bit (Bus Lock Interrupt Enable)**

When the BLIE and the CFDC0ERFL.BLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_RESET mode.

**ALIE bit (Arbitration Lost Interrupt Enable)**

When the ALIE and the CFDC0ERFL.ALF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_RESET mode.

**TAIE bit (Transmission Abort Interrupt Enable)**

When the TAIE bit is 1 and a transmission is successfully aborted from a TX MB belonging to the corresponding CAN channel, an interrupt request is generated.

Do not write to this bit in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_RESET mode.

**EOCOIE bit (Error Occurrence Counter Overflow Interrupt Enable)**

When the EOCOIE bit is 1 and the CFDC0FDSTS.EOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The EOCOIE bit cannot be set in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_RESET mode.

**SOCOIE bit (Successful Occurrence Counter Overflow Interrupt Enable)**

When the SOCOIE bit is 1 and the CFDC0FDSTS.SOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The SOCOIE bit cannot be set in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_RESET mode.

**TDCVFIE bit (Transceiver Delay Compensation Violation Interrupt Enable)**

When the TDCVFIE bit is 1 and the CFDC0FDSTS.TDCVF bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The TDCVFIE bit cannot be set in CH\_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH\_RESET mode. Do not set this bit when in Classical CAN mode.

Note: This bit is not available in the classical CAN function.

**BOM[1:0] bits (Channel Bus-Off Mode)**

The BOM[1:0]bits control the timing of the recovery from Bus-Off mode of the CANFD Channel.

Do not write to these bits in CH\_SLEEP mode. Only write to these bits when the related CANFD channel is in CH\_RESET mode.

Only write to these bits when the related CANFD channel is in CH\_RESET mode.

**ERRD bit (Channel Error Display)**

The ERRD bit controls the display mode of the error flag bits [14:8] in the Channel Error Flag Register (CFDC0ERFL).

If the ERRD bit is 0 and more than one error occur at the same time, the error flag bits are set for all the errors that occurred at the same time. No further errors are flagged until CFDC0ERFL[14:8] is cleared.

Do not write to the ERRD bit in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

**CTME bit (Channel Test Mode Enable)**

The CTME bit enables the channel test modes.

Do not write to this bit in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_HALT mode.

### **CTMS[1:0] bits (Channel Test Mode Select)**

The CTMS[1:0] bits are used to select the required test mode.

Do not write to these bits in CH\_SLEEP or CH\_RESET mode. Only write to these bits when the related CANFD channel is in CH\_HALT mode.

These bits are cleared automatically when the related CANFD channel is in CH\_RESET mode.

### **BFT bit (Bit Flip Test)**

The BFT bit checks the internal CRC generator logic of the protocol controller.

It inverts the first bit (ID bit) of the CAN message data stream being received, so that the internal generated CRC result will not match the received CRC value of the frame. Refer to the bit stuffing rule, when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The internal generated CRC value is always observed in the following registers:

- CFDC0ERFL.CRCREG (Classical CAN frames)
- CFDC0FDCRC.CRCREG (CANFD frames).<sup>\*1</sup>

Note 1. This feature is not available in the classical CAN function.

Some restriction exist when using this bit:

Other CAN node will send a reference message and the receiver node(s) can invert one bit of incoming bit stream.

Note: The transmitter and receiver modes share the same CRC generator, therefore it is not necessary to consider the modes separately when testing.

The Bit Flip test mode is enabled if the BFT (new control signal that inverts the first bit of the bit stream) and CTME bits are both 1 and CFDC0CTR.CTMS is 0x00.

If this function is used by a transmitting node, a bit error or an arbitration lost will occur.

Do not write to the BFT bit in CH\_SLEEP mode. Users should not use this function when the Self test mode 1 (Internal Loop back mode). Only write to this bit when the related CANFD channel is in CH\_HALT mode.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode.

### **ROM bit (Restricted Operation Mode)**

When the ROM and CTME bits are both 1, the restricted operation mode is enabled. This mode should only be used in basic test mode (CFDC0CTR.CTMS[1:0] = 00b).

The ROM bit cannot be set in CH\_SLEEP mode. Only write to this bit when the related CANFD channel is in CH\_HALT mode.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode. Do not set this bit when in Classical CAN mode.

Note: This bit is not available in the classical CAN function.

### 32.2.4 CFDC0STS : Status Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TEC[7:0]								REC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ESIF	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
0	CRSTSTS	Channel Reset Status 0: Channel not in Reset mode 1: Channel in Reset mode	R
1	CHLTSTS	Channel Halt Status 0: Channel not in Halt mode 1: Channel in Halt mode	R
2	CSLPSTS	Channel Sleep Status 0: Channel not in Sleep mode 1: Channel in Sleep mode	R
3	EPSTS	Channel Error Passive Status 0: Channel not in error passive state 1: Channel in error passive state	R
4	BOSTS	Channel Bus-Off Status 0: Channel not in bus-off state 1: Channel in bus-off state	R
5	TRMSTS	Channel Transmit Status 0: Channel is not transmitting 1: Channel is transmitting	R
6	RECSTS	Channel Receive Status 0: Channel is not receiving 1: Channel is receiving	R
7	COMSTS	Channel Communication Status 0: Channel is not ready for communication 1: Channel is ready for communication	R
8	ESIF <sup>*1</sup>	Error State Indication Flag 0: No CANFD message has been received when the ESI flag was set 1: At least one CANFD message was received when the ESI flag was set	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
23:16	REC[7:0]	Reception Error Count These bits increment or decrement the counter value according to error status of the CAN channel during reception.	R
31:24	TEC[7:0]	Transmission Error Count These bits increment or decrement the counter value according to error status of the CAN channel during transmission.	R

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is not available in the classical CAN function.

Channel Status Register shows the mode, error and transmission or reception status of the related channel together with its reception and transmission error count values.

**CRSTSTS bit (Channel Reset Status)**

The CRSTSTS bit indicates whether the related CAN channel is in Reset mode.

This bit is set automatically when the related CAN channel enters Channel Reset mode. When the mode is changed from Reset mode to Sleep mode, the CRSTSTS bit remains 1.

This bit is cleared automatically when the related CAN channel exits the Channel Reset mode, except when changing to Sleep mode.

**CHLTSTS bit (Channel Halt Status)**

The CHLTSTS bit indicates whether the related CAN channel is in Halt mode.

This bit is set automatically when the related CAN module enters Halt mode, and is cleared automatically when the related CAN module exits Halt mode.

**CSLPSTS bit (Channel Sleep Status)**

The CSLPSTS bit indicates whether the related CAN channel is in Sleep mode.

This bit is set automatically when the related CANFD channel enters Sleep mode, and is cleared automatically when the related CANFD channel exits Sleep mode.

**EPSTS bit (Channel Error Passive Status)**

The EPSTS bit indicates whether the related CANFD channel has entered the error passive state.

This bit is set automatically when the value of the CAN Transmission or Reception Counter Register exceeds the value of 0x7F.

This bit is cleared automatically when the related CANFD channel exits the error passive state or enters Reset mode.

**BOSTS bit (Channel Bus-Off Status)**

The BOSTS bit indicates whether the related CANFD channel has entered the error bus-off state.

This bit is set automatically when the value of the related CAN Transmission Error Count Register exceeds 0xFF and the related CANFD channel is in the bus-off state (CAN Transmission Error Count Register > 0xFF).

This bit is cleared automatically when the related CANFD channel exits bus-off state.

**TRMSTS bit (Channel Transmit Status)**

The TRMSTS bit indicates whether the related CANFD channel is transmitting a message.

This bit is set automatically when the related CANFD channel is operating as a transmitter node or is in the bus-off state.

This bit is cleared automatically when the related CANFD channel is in the bus-idle state or starts operating as a receiver node.

**RECSTS bit (Channel Receive Status)**

The RECSTS bit indicates whether the related CANFD channel is receiving a message.

This bit is set automatically when the related CANFD channel is operating as a receiver node.

This bit is cleared automatically when the related CANFD channel is in the bus-idle state or starts operating as a transmitter node.

**COMSTS bit (Channel Communication Status)**

The COMSTS bit indicates whether the related CANFD channel is ready for communication.

This bit is set automatically when the related CANFD channel is ready to perform communication following the detection of 11 consecutive recessive bits after exiting the Reset or Halt mode.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET or CD\_HALT mode.

Note: This bit is 1 during bus-off state.

**ESIF bit (Error State Indication Flag)**

The ESIF bit is set when the ESI bit is sampled recessively for a reception CAN message without any error. When in Loopback or Mirror mode, the self-transmitted messages are considered reception messages.

If a set from the CANFD channel occurs simultaneously with a clear by a write access, then the bit is set.

This bit is cleared by writing 0 to it. This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Note: This bit is not available in the classical CAN function.

**REC[7:0] bits (Reception Error Count)**

The REC[7:0] bits increment or decrement the counter value according to error status of the CANFD channel during reception, and display the value of the REC error counter.

The value in bus-off state is indeterminate.

These bits are cleared automatically when the CANFD module enters GL\_RESET or the CANFD channel is in CH\_RESET mode.

**TEC[7:0] bits (Transmission Error Count)**

The TEC[7:0] bits increment or decrement the counter value according to error status of the CANFD channel during transmission, and display the value of the TEC error counter.

Only write to these bits when in test mode and CANFD channel is in CH\_HALT mode.

These bits are cleared automatically when CANFD module is in GL\_RESET or CANFD channel is in CH\_RESET mode.

**32.2.5 CFDC0ERFL : Error Flag Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x000C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CRCREG[14:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ADER R	B0ER R	B1ER R	CERR	AERR	FERR	SERR	ALF	BLF	OVLf	BORF	BOEF	EPF	EWf	BEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BEF	Bus Error Flag 0: Channel bus error not detected 1: Channel bus error detected	R/W
1	EWf	Error Warning Flag 0: Channel error warning not detected 1: Channel error warning detected	R/W
2	EPF	Error Passive Flag 0: Channel error passive not detected 1: Channel error passive detected	R/W
3	BOEF	Bus-Off Entry Flag 0: Channel bus-off entry not detected 1: Channel bus-off entry detected	R/W
4	BORF	Bus-Off Recovery Flag 0: Channel bus-off recovery not detected 1: Channel bus-off recovery detected	R/W



Bit	Symbol	Function	R/W
5	OVLFL	Overload Flag 0: Channel overload not detected 1: Channel overload detected	R/W
6	BLFL	Bus Lock Flag 0: Channel bus lock not detected 1: Channel bus lock detected	R/W
7	ALFL	Arbitration Lost Flag 0: Channel arbitration lost not detected 1: Channel arbitration lost detected	R/W
8	SERR	Stuff Error 0: Channel stuff error not detected 1: Channel stuff error detected	R/W
9	FERR	Form Error 0: Channel form error not detected 1: Channel form error detected	R/W
10	AERR	Acknowledge Error 0: Channel acknowledge error not detected 1: Channel acknowledge error detected	R/W
11	CERR	CRC Error 0: Channel CRC error not detected 1: Channel CRC error detected	R/W
12	B1ERR	Bit 1 Error 0: Channel bit 1 error not detected 1: Channel bit 1 error detected	R/W
13	B0ERR	Bit 0 Error 0: Channel bit 0 error not detected 1: Channel bit 0 error detected	R/W
14	ADERR	Acknowledge Delimiter Error 0: Channel acknowledge delimiter error not detected 1: Channel acknowledge delimiter error detected	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
30:16	CRCREG[14:0]	CRC Register value These bits show the CRC value calculated for the CAN2.0 CAN frame.	R
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Channel Error Flag register shows the status of various error conditions detectable regardless of the setting of the related CAN Channel Error Interrupt Enable Register. It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) to check when each error condition occurs.

For this register, only a single bit can be cleared by software. Do not use the bit clear instruction to clear the bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Example in assembler language to clear the CFDC0ERFL.BEF bit:

```
mov.b #0x0FE, CFDC0ERFL ;
```

### BEF bit (Bus Error Flag)

The BEF bit indicates a detection of a CAN channel bus error state, flagged by bits [14:8] in this register.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when a bus error is detected, and is cleared automatically when the related CANFD channel is in CH\_RESET mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

**EWf bit (Error Warning Flag)**

The EWf bit indicates whether an error warning condition has been detected for the CAN channel.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when either TEC or REC exceeds 0x5F.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x5F. Therefore, if the TEC or REC remains > 0x5F and the EWf bit is cleared by software, it is not set again until both the TEC and REC go below 0x60 and either TEC or REC crosses over again from a value 0x5F to a value > 0x5F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH\_RESET mode.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

**EPF bit (Error Passive Flag)**

The EPF bit indicates a detection of a CAN channel error passive state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state becomes error passive state.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x7F. Therefore, if the TEC or REC remains > 0x7F and the bit is cleared by software, it is not set again until both the TEC and REC go below 0x80 and either TEC or REC crosses over again from a value ≤ 0x7F to a value > 0x7F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH\_RESET mode.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

**BOEF bit (Bus-Off Entry Flag)**

The BOEF bit indicates a detection of a CAN channel bus-off entry state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state enters the bus-off state.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode. If a set condition occurs simultaneously with a clear condition, then the bit is set.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

**BORF bit (Bus-Off Recovery Flag)**

The BORF bit indicates a detection of a CAN channel bus-off recovery state.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically if CAN channel recovers from bus-off state in the following conditions:

- When CFDC0CTR.BOM is 00b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs
- When CFDC0CTR.BOM is 10b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs
- When CFDC0CTR.BOM is 11b and normal recovery (11 consecutive recessive bits x 128 times detected) occurs.

The bit is not set if CAN channel recovers from bus-off state in the following conditions:

- When CAN Reset mode is requested
- When CFDC0CTR.RTBO is set to 1 (the CAN channel returns to error active)
- When CFDC0CTR.BOM is 01b
- When CFDC0CTR.BOM is 11b and a halt request is asserted before the CAN channel reaches the end of the bus-off state.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode. If a set condition occurs simultaneously with a clear condition, the flag is set.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

**OVLf bit (Overload Flag)**

The OVLf flag indicates a detection of a CAN channel overload state.

The OVLf bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when an overload condition is detected. If a set condition occurs simultaneously with a clear condition, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

**BLF bit (Bus Lock Flag)**

The BLF bit indicates a detection of a CAN channel bus lock condition.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

This bit is set automatically when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH\_RESET mode.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

It is cleared automatically when the related CANFD channel is in CH\_RESET mode.

**ALF bit (Arbitration Lost Flag)**

The ALF bit indicates a detection of a CAN channel bus arbitration lost condition.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

The bit is set automatically when an arbitration lost condition is detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CANFD channel is in CH\_RESET mode.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

**SERR bit (Stuff Error)**

The SERR bit indicates a detection of a CAN stuff error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a stuff error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

**FERR bit (Form Error)**

The FERR bit indicates a detection of a CAN form error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.

3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

#### **AERR bit (Acknowledge Error)**

The AERR bit indicates a detection of a CAN acknowledge error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when an acknowledge error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION.

#### **CERR bit (CRC Error)**

The CERR bit indicates a detection of a CAN CRC error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a CRC error is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

#### **B1ERR bit (Bit 1 Error)**

The B1ERR bit indicates a detection of a recessive bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a recessive bit error (expected recessive bit, sampled as dominant bit) is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

#### **B0ERR bit (Bit 0 Error)**

The B0ERR bit indicates a detection of a dominant bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a dominant bit error (expected dominant bit, sampled as recessive bit) is detected. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

#### **ADERR bit (Acknowledge Delimiter Error)**

The ADERR bit indicates a detection of an acknowledge delimiter bit error.

This bit is cleared by writing 0 to it, and can only be set by CANFD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected during the acknowledge delimiter state of frame transmission. If CFDC0CTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode. If CFDC0CTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at CFDC0ERFL[14:8] is already set. Otherwise, this bit is set if CFDC0ERFL[14:8] is 0000000b.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

#### **CRCREG[14:0] bits (CRC Register value)**

The CRCREG[14:0] bits read the calculated CRC value when CFDC0CTR.CTME bit is 1 for the channel.

If CFDC0CTR.CTME bit is 0, then these bits are always read as 0.

These bits show the CAN2.0 CRC value calculated by the CANFD channel logic when the CTME bit is enabled.

The CFDC0ERFL.CRCREG value is updated in the first bit of the CRC field of the CAN frame (reception and transmission).

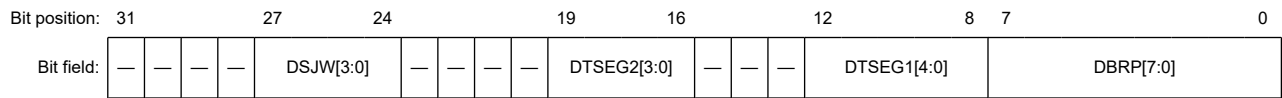
These bits are cleared automatically when the related CANFD channel is in CH\_RESET mode.

### **32.2.6 CFDC0DCFG : Data Bitrate Configuration Register**

This register is not available in the classical CAN function.

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0100



Value after reset: 0

Bit	Symbol	Function	R/W
7:0	DBRP[7:0]	Channel Data Baud Rate Prescaler Data Baud Rate Prescaler division ratio	R/W
12:8	DTSEG1[4:0]	Timing Segment 1 0x00: Reserved 0x01: 2 Tq 0x02: 3 Tq 0x03: 4 Tq ⋮ 0x1E: 31 Tq 0x1F: 32 Tq	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
19:16	DTSEG2[3:0]	Timing Segment 2 0x0: Reserved 0x1: 2 Tq ⋮ 0xE: 15 Tq 0xF: 16 Tq	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	DSJW[3:0]	Resynchronization Jump Width 0x0: 1 Tq 0x1: 2 Tq ⋮ 0xF: 16 Tq	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3  
 Note: Tq means time quantum.

The Data Bitrate Configuration Register configures the transmission/reception data baud rate parameters of the channels. The channel of Classical CAN mode does not perform configuration of this register.

### DBRP[7:0] bits (Channel Data Baud Rate Prescaler)

The DBRP[7:0] bits define the peripheral bus clock periods contained in a time quantum.

Do not write to these bits in CH\_OPERATION or CH\_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

### DTSEG1[4:0] bits (Timing Segment 1)

The DTSEG1[4:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. A value from 2 to 32 time quanta can be set.

The DTSEG1[4:0] bits are also used to set the propagation segment.

Do not write to these bits in CH\_OPERATION or CH\_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode. Do not write any other value to these bits. See [section 32.4.1.2. CAN Bit Timing](#) for more details.

### DTSEG2[3:0] bits (Timing Segment 2)

The DTSEG2[3:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. A value from 2 to 16 time quanta can be set.

Do not write to these bits in CH\_OPERATION or CH\_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode. Do not write any other value to these bits.

**DSJW[3:0] bits (Resynchronization Jump Width)**

The DSJW[3:0] bits set the synchronization jump width. A value from 1 to 16 time quanta can be set.

Do not write to these bits in CH\_OPERATION or CH\_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

**32.2.7 CFDC0FDCFG : CANFD Configuration Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CLOE	REFE	FDOE	—	—	—	—	TDCO[7:0]							
Value after reset:	0	0/1 <sup>1</sup>	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ESIC	TDCE	TDCOC	—	—	—	—	—	EOCCFG[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	EOCCFG[2:0]	Error Occurrence Counter Configuration 0 0 0: All transmitter or receiver CAN frames 0 0 1: All transmitter CAN frames 0 1 0: All receiver CAN frames 0 1 1: Reserved 1 0 0: Only transmitter or receiver CANFD data-phase (fast bits) 1 0 1: Only transmitter CANFD data-phase (fast bits) 1 1 0: Only receiver CANFD data-phase (fast bits) 1 1 1: Reserved	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	TDCOC <sup>*2</sup>	Transceiver Delay Compensation Offset Configuration 0: Measured + offset 1: Offset-only	R/W
9	TDCE <sup>*2</sup>	Transceiver Delay Compensation Enable 0: Transceiver delay compensation disabled 1: Transceiver delay compensation enabled	R/W
10	ESIC <sup>*2</sup>	Error State Indication Configuration 0: The ESI bit in the frame represents the error state of the node itself 1: The ESI bit in the frame represents the error state of the message buffer if the node itself is not in error passive. If the node is in error passive, then the ESI bit is driven by the node itself.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TDCO[7:0] <sup>*2</sup>	Transceiver Delay Compensation Offset	R/W
27:24	—	These bits are read as 0. The write value should be 0.	R/W
28	FDOE <sup>*2</sup>	FD-Only Enable 0: FD-only mode disabled 1: FD-only mode enabled	R/W

Bit	Symbol	Function	R/W
29	REFE	RX Edge Filter Enable 0: RX edge filter disabled 1: RX edge filter enabled	R/W
30	CLOE <sup>*2 *3</sup>	Classical CAN Enable 0: Classical CAN mode disabled 1: Classical CAN mode enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. The value after reset is 0 for products that support the CAN-FD protocol, and 1 for products that support only classical CAN protocol.

Note 2. These bits are not available in the classical CAN function.

Note 3. This bit can only be written for products that support the CAN-FD protocol. For products that support only classical CAN protocol, this bit is reserved and fixed to 1.

The CANFD Configuration Register configures which communication direction (transmitter/receiver) errors are counted.

### **ECCCFG[2:0] bits (Error Occurrence Counter Configuration)**

The ECCCFG[2:0] bits select which type of CAN frame configuration and direction, including protocol errors are counted.

Do not write to these bits in CH\_OPERATION or CH\_SLEEP mode.

Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

### **TDCOC bit (Transceiver Delay Compensation Offset Configuration)\*1**

The TDCOC bit selects which offset is used when defining the position of the secondary sample point (SSP) for the CANFD channel. If the bit is set to 0, the position of the SSP is the measured transceiver delay plus the fixed offset. If the bit is 1, the position of the SSP is defined only by the offset.

Do not write to this bit in CH\_OPERATION or CH\_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH\_RESET or CH\_HALT mode. Do not set this bit when in Classical CAN mode.

### **TDCE bit (Transceiver Delay Compensation Enable)\*1**

The TDCE bit enables the transceiver delay compensation for the CANFD channel.

Do not write to this bit in CH\_OPERATION or CH\_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH\_RESET or CH\_HALT mode. Do not set this bit when in Classical CAN mode.

### **ESIC bit (Error State Indication Configuration)\*1**

The ESIC bit controls the transmission of either the ESI flag information or the message of ESI flag information (CFDCFFDCSTS.CFESI or CFDTMFDCSTRb.TMESI).

Do not write to this bit in CH\_OPERATION or CH\_SLEEP mode.

Only write to this bit when the related CANFD channel is in CH\_RESET or CH\_HALT mode. Do not set this bit when in Classical CAN mode.

### **TDCO[7:0] bits (Transceiver Delay Compensation Offset)\*1**

The TDCO[7:0] bits set the secondary sample point offset. How this value is used, depends on the CFDC0FDCFG.TDCOC setting.

If CFDC0FDCFG.TDCOC = 0, the transceiver delay compensation result is equal to the Trv\_Delay (measured delay) + the value in CFDC0FDCFG.TDCO, rounded down to the nearest integer number of time quanta. Otherwise, the result is equal to the value in CFDC0FDCFG.TDCO. See [section 32.4.1.5. Transmitter Delay Compensation](#) for details on how CFDC0FDCFG.TDCO is used.

The actual offset value is interpreted as TDCO + 1. For example, if 4 is set in TDCO, the offset is 5 clock cycles. Clock cycle is 1 cycle of CAN channel DLL clock.

Do not write to the TDCO[7:0] bits in CH\_OPERATION or CH\_SLEEP mode.



Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode. Do not set this bit when in Classical CAN mode.

**FDOE bit (FD-Only Enable)\*1**

The FDOE bit enables the reception and transmission of CANFD-only frames. If enabled, communication in Classical CAN frame format is disabled. Transmission of Classical CAN frames is not possible because the FDF bit of the message buffer is a don't care (CFDCFFDCSTS.CFFDF/CFDTMFDCTRb.TMFDF).

If messages with Classical CAN frame format are received, the protocol controller treats them as invalid frames and response with error frames. When a Classical CAN frame is configured for transmitting, the FDF bit is sent as recessive, therefore an FD frame is sent. If the data length code (DLC) is configured of greater than 8 bytes, the remaining data bytes are padded with 0xCC.

The FDOE bit cannot be written in CH\_OPERATION, CH\_HALT or CH\_SLEEP mode.

Do not set CFDC0FDCFG.FDOE and CFDC0FDCFG.CLOE simultaneously.

**REFE bit (RX Edge Filter Enable)**

The REFE bit enables the RX edge filter during the IDLE detection (bus integration). When the bit is enabled, two consecutive dominant time quanta are required to detect a synchronization edge.

The REFE bit cannot be written in CH\_OPERATION, CH\_HALT and CH\_SLEEP mode. Do not set this bit when in Classical CAN mode.

**CLOE bit (Classical CAN Enable)\*1**

The CLOE bit enables the Classical CAN mode. If this bit is 1, the protocol controller can only send classical frames and response with a form or CRC error on FD frames.

Do not set CFDC0FDCFG.CLOE and CFDC0FDCFG.FDOE simultaneously.

CFDC0FDCFG.CLOE	CFDC0FDCFG.FDOE	Channel mode
0	0	CANFD mode
0	1	FD-only mode
1	0	Classical CAN mode
1	1	Reserved

The CANFD mode is available only for CANFD supported product.

Do not write to this bit in CH\_OPERATION, CH\_HALT or CH\_SLEEP mode.

Only write to these bits when the CANFD channel is in CH\_RESET mode.

Note 1. These bits are not available in the classical CAN function.

**32.2.8 CFDC0FDCTR : CANFD Control Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOCC LR	EOCC LR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EOCCLR	Error Occurrence Counter Clear 0: No error occurrence counter clear 1: Clear error occurrence counter	R/W
1	SOCCLR	Successful Occurrence Counter Clear 0: No successful occurrence counter clear 1: Clear successful occurrence counter	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The CANFD Control Register controls the error and successful occurrence counters.

### EOCCLR bit (Error Occurrence Counter Clear)

The EOCCLR bit is used to clear the error occurrence counter.

Do not write to this bit in CH\_SLEEP or CH\_RESET mode. The read value is always 0.

This bit is cleared automatically by the CANFD module logic and when the related CANFD channel is in CH\_RESET mode.

### SOCCLR bit (Successful Occurrence Counter Clear)

The SOCCLR bit is used to clear the successful occurrence counter.

Do not write to this bit in CH\_SLEEP or CH\_RESET mode. The read value is always 0.

This bit is cleared automatically by the CANFD module logic and when the related CANFD channel is in CH\_RESET mode.

## 32.2.9 CFDC0FDSTS : CANFD Status Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x010C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SOC[7:0]							EOC[7:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TDCV F	—	—	—	—	—	SOCO	EOCO	TDCR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TDCR[7:0] <sup>*1</sup>	Transceiver Delay Compensation Result	R
8	EOCO	Error Occurrence Counter Overflow 0: Error occurrence counter has not overflowed 1: Error occurrence counter has overflowed	R/W
9	SOCO	Successful Occurrence Counter Overflow 0: Successful occurrence counter has not overflowed 1: Successful occurrence counter has overflowed	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	TDCVF <sup>*1</sup>	Transceiver Delay Compensation Violation Flag 0: Transceiver delay compensation violation has not occurred 1: Transceiver delay compensation violation has occurred	R/W
23:16	EOC[7:0]	Error Occurrence Counter These bits show the error occurrence counter value.	R

Bit	Symbol	Function	R/W
31:24	SOC[7:0]	Successful occurrence counter These bits show the successful occurrence counter value.	R

Note: S-TYPE-3, P-TYPE-3

Note 1. These bits are not available in the classical CAN function.

The CANFD Status Register indicates the transceiver compensation delay result and its related FIFO message lost status.

### TDCR[7:0] bits (Transceiver Delay Compensation Result)

The TDCR[7:0] bits are set when the transceiver delay has been measured.

The measured delay is a multiple of the CAN channel DLL clock. The result depends on the CFDC0FDCFG.TDCOC configuration and the offset value in CFDC0FDCFG.TDCO. See [section 32.4.1.5. Transmitter Delay Compensation](#) for details on how this value is derived.

The TDCR[7:0] bits are updated at the falling edge between FDF and the RES bit when CFDC0FDCFG.TDCOC = 0 and the transceiver delay compensation is enabled (CFDC0FDCFG.TDCE = 1).

These bits are cleared automatically when the related CANFD channel is in CH\_RESET mode.

Note: These bits are not available in the classical CAN function.

### EOCO bit (Error Occurrence Counter Overflow)

The EOCO bit indicates whether the related CAN channel error occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDC0FDSTS.EOC is 0xFF and a CAN bus error is detected based on the configuration defined in CFDC0FDCFG.EOCCFG.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

### SOCO bit (Successful Occurrence Counter Overflow)

The SOCO bit indicates whether the related CAN channel successful occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDC0FDSTS.SOC is 0xFF and a successful message reception or successful message transmission occurs.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode.

Write to this bit only when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

### TDCVF bit (Transceiver Delay Compensation Violation Flag)

The CANFD module captures internally the transmitted data bit-by-bit. This data is then compared against the received CAN bus level which is delayed by the transceiver loop delay.

The transceiver delay has some variations depending on the physical parameters such as temperature. The result bit CFDC0FDSTS.TDCR is updated by each message. However, temporary maximum delay violation could be missed. Therefore, the TDCVF bit captures this violation.

This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when the transceiver delay compensation is greater than the maximum delay compensation (6 data bit times - 2 clk\_dlc) and the internal bit is overrun.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Note: This bit is not available in the classical CAN function.

**EOC[7:0] bits (Error Occurrence Counter)**

The EOC[7:0] bits are used together with the SOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

This higher error rate can be detected depending on the configuration of the CFDC0FDCFG.EOCCFG bits.

The EOC[7:0] bits are set only by CANFD module logic. These bits are cleared by writing 1 to CFDC0FDCTR.EOCCLR. Writing any other value has no effect.

These bits are updated when an error occurs, according to the configuration of the CFDC0FDCFG.EOCCFG bits. When the counter reaches the value of 0xFF, the update stops.

These bits are cleared automatically when the related CANFD channel is in CH\_RESET mode.

**SOC[7:0] bits (Successful occurrence counter)**

The SOC[7:0] bits are used together with the EOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

The SOC[7:0] bits are set only by CANFD module logic. Writing any other value has no effect.

These bits are updated when the occurrence of any error-free messages on the bus is detected through reception or transmission. When the counter reaches the value of 0xFF, the update stops.

Note: In Loopback mode, the counter is incremented twice.

These bits are cleared by writing 1 to CFDC0FDCTR.SOCCLR.

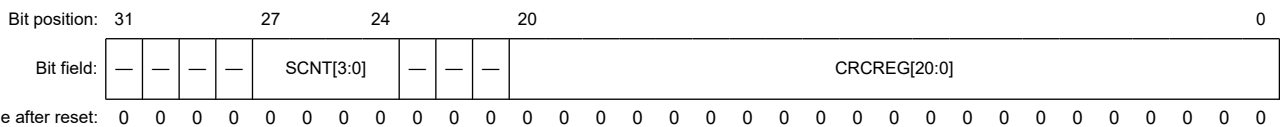
These bits are cleared automatically when the related CANFD channel is in CH\_RESET mode.

**32.2.10 CFDC0FDCRC : CANFD CRC Register**

This register is not available in the classical CAN function.

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0110



Bit	Symbol	Function	R/W
20:0	CRCREG[20:0]	CRC Register value These bits show the CRC value calculated for the CANFD frame.	R
23:21	—	These bits are read as 0.	R
27:24	SCNT[3:0]	Stuff bit count These bits shows the stuff bit count (mod 8) for the CANFD frame.	R
31:28	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The CANFD CRC Register holds the CRC value calculated for the CANFD frame.

**CRCREG[20:0] bits (CRC Register value)**

The CRCREG[20:0] bits contain the CRC value calculated by the CANFD channel logic when the CFDC0CTR.CTME bit is enabled.

The CFDC0FDCRC.CRCREG value is updated in the first bit of the CRC field of the CANFD frame (reception and transmission).

When the CFDC0CTR.CTME bit is 0, the CRCREG[20:0] bits are always read as 0.

When bit 17th of the CRC field is used, CRCREG[20:17] are always read as 0.

These bits are cleared automatically when the related CANFD channel is in CH\_RESET mode.

**SCNT[3:0] bits (Stuff bit count)**

The SCNT[3:0] bits contain the stuff count value of the CANFD frame. These bits indicate the number of inserted stuff bits (modulo 8, Graycoded) for a CANFD frame when the CFDC0CTR.CTME bit is enabled in CFDC0FDCRC.SCNT[3:1]. SCNT[0] is the parity bit.

When the CFDC0CTR.CTME bit is 0, the SCNT[3:0] bits are always read as 0.

The SCNT value is updated in the first bit of CRC field of the CANFD frame (reception and transmission).

These bits are cleared automatically when the related CANFD channel is in CH\_RESET mode.

**32.2.11 CFDGCFG : Global Configuration Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ITRCP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TSSS	TSP[3:0]			—	—	CMPO C	DCS	MME	DRE	DCE	TPRI	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TPRI	Transmission Priority 0: ID priority 1: Message buffer number priority	R/W
1	DCE	DLC Check Enable 0: DLC check disabled 1: DLC check enabled	R/W
2	DRE	DLC Replacement Enable 0: DLC replacement disabled 1: DLC replacement enabled	R/W
3	MME	Mirror Mode Enable 0: Mirror mode disabled 1: Mirror mode enabled	R/W
4	DCS	Data Link Controller Clock Select 0: CANFD core clock (CANFDCLK) 1: External oscillator clock (CANMCLK)	R/W
5	CMPOC <sup>*1</sup>	CANFD Message Payload Overflow Configuration 0: Message is rejected 1: Message payload is cut to fit to configured message size	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
11:8	TSP[3:0]	Timestamp Prescaler 0x0: Timestamp prescaler = 1 0x1: Timestamp prescaler = 2 0x2: Timestamp prescaler = 4 0x3: Timestamp prescaler = 8 ⋮ 0xD: Timestamp prescaler = 8192 0xE: Timestamp prescaler = 16384 0xF: Timestamp prescaler = 32768	R/W
12	TSSS	Timestamp Source Select 0: Source clock for timestamp counter is peripheral clock 1: Source clock for timestamp counter is bit time clock	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
31:16	ITRCP[15:0]	Interval Timer Reference Clock Prescaler FIFO interval timer prescaler value	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is not available in the classical CAN function.

The Global Configuration Register is used to select the transmission priority to be used for all the TX message buffers and the clock source for the CAN protocol engine of CAN channel. The CFDGCFG register is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

#### TPRI bit (Transmission Priority)

The TPRI bit selects the transmission priority for CAN channel.

Do not write to this bit in GL\_SLEEP mode. Only write to this bit when CANFD module is in GL\_RESET mode.

Message buffer number priority should not be used together with TX queue transmission.

#### DCE bit (DLC Check Enable)

The DCE bit enables data length code (DLC) check for CAN channel.

Do not write to this bit in GL\_SLEEP mode. Only write to this bit when CANFD module is in GL\_RESET mode.

#### DRE bit (DLC Replacement Enable)

When the DRE bit is 1 and the DCE is 1, the CANFD stores the configured value (CFDGAFLP0r.GAFLDLC) of the DLC in the destination RX message buffer or FIFO buffer if the DLC check passes. Otherwise, the DLC value in the destination RX message buffer or FIFO buffer is unchanged.

Do not write to this bit in GL\_SLEEP mode. Only write to this bit when CANFD module is in GL\_RESET mode.

#### MME bit (Mirror Mode Enable)

The MME bit enables the Mirror mode for CAN channel.

Do not write to this bit in GL\_SLEEP mode. Only write to this bit when CANFD module is in GL\_RESET mode.

#### DCS bit (Data Link Controller Clock Select)

The DCS bit selects CANFDCLK or CANMCLK as the clock source for CAN communication.

Do not write to this bit in GL\_SLEEP or GL\_OPERATION mode. Only write to this bit when CANFD module is in GL\_RESET mode.

#### CMPOC bit (CANFD Message Payload Overflow Configuration)

The CMPOC bit controls the message payload acceptance mechanism when the received payload is higher than the message buffer payload size CFDRMNB.RMPLS, CFDRFCCa.RFPLS, and CFDCFCC.CFPLS. The received message payload is always compared with the available message payload size in the message buffer.

Do not write to this bit in GL\_SLEEP or GL\_OPERATION mode. Only write to this bit when CANFD module is in GL\_RESET mode.

When this bit is set and payload overflow occurs, the DLC value is stored in the RX message buffer or FIFO buffer unchanged.

Note: This bit is not available in the classical CAN function.

**TSP[3:0] bits (Timestamp Prescaler)**

The value configured in the TSP[3:0] bits defines the period of the clock source used for the timestamp counter. Do not write to this bit in GL\_SLEEP mode. Only write to this bit when CANFD module is in GL\_RESET mode.

**TSSS bit (Timestamp Source Select)**

The TSSS bit allows the selection of the clock source for the timestamp counter. Do not write to this bit in GL\_SLEEP mode. Only write to this bit when CANFD module is in GL\_RESET mode. Additionally, do not set this bit to 1 when CANFD communication is used.\*1

Note: The bit time clock varies depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

**ITRCP[15:0] bits (Interval Timer Reference Clock Prescaler)**

The ITRCP[15:0] bits allow the definition of a reference clock for the FIFO interval timer source clock. When these bits are 0x0000, the timer is disabled. Do not write to this bit in GL\_SLEEP mode. Only write to this bit when CANFD module is in GL\_RESET mode.

**32.2.12 CFGDCTR : Global Control Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CMPOFIE	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0 1

Bit	Symbol	Function	R/W
1:0	GMDC[1:0]	Global Mode Control 0 0: Global operation mode request 0 1: Global reset mode request 1 0: Global halt mode request 1 1: Keep current value	R/W
2	GSLPR	Global Sleep Request 0: Global sleep request disabled 1: Global sleep request enabled	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	DEIE	DLC Check Interrupt Enable 0: DLC check interrupt disabled 1: DLC check interrupt enabled	R/W
9	MEIE	Message Lost Error Interrupt Enable 0: Message lost error interrupt disabled 1: Message lost error interrupt enabled	R/W
10	THLEIE	TX History List Entry Lost Interrupt Enable 0: TX history list entry lost interrupt disabled 1: TX history list entry lost interrupt enabled	R/W

Bit	Symbol	Function	R/W
11	CMPOFIE <sup>*1</sup>	CANFD Message Payload Overflow Flag Interrupt Enable 0: CANFD message payload overflow flag interrupt disabled 1: CANFD message payload overflow flag interrupt enabled	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	TSRST	Timestamp Reset 0: Timestamp not reset 1: Timestamp reset	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is not available in the classical CAN function.

The Global Control Register controls the global mode of the CANFD module and the timestamp function. The register also enables and disables the global error interrupts.

### GMDC bits (Global Mode Control)

The GMDC bits can be used to configure the modes for the CANFD module. Additionally, if `CFDGCTR.GSLPR` bit is 1 when the CANFD module is in Reset mode, the CANFD module enters Global Sleep mode.

Setting the GMDC bits to 11b has no effect. Mode transition is described in detail in [section 32.3.2. Global Modes](#).

Do not write to this bit when the CANFD module is in `GL_SLEEP` mode.

### GSLPR bit (Global Sleep Request)

The `GSLPR` bit globally selects the sleep request for CANFD module including CAN channels. Channel sleep request is set automatically for channels.

Only write to this bit when the CANFD module is in `GL_RESET` or `GL_SLEEP` mode.

### DEIE bit (DLC Check Interrupt Enable)

When the `DEIE` bit is 1, an interrupt is generated if a DLC error is detected in the received frames.

Do not write to this bit when the CANFD module is in `GL_SLEEP` mode.

### MEIE bit (Message Lost Error Interrupt Enable)

When the `MEIE` bit is 1, an interrupt is generated if a message lost condition occurs.

Do not write to this bit when the CANFD module is in `GL_SLEEP` mode.

### THLEIE bit (TX History List Entry Lost Interrupt Enable)

When the `THLEIE` bit is 1, an interrupt is generated if a TX history list entry lost condition occurs.

Do not write to this bit when the CANFD module is in `GL_SLEEP` mode.

### CMPOFIE bit (CANFD Message Payload Overflow Flag Interrupt Enable)

When the `CMPOFIE` bit is 1, an interrupt is generated when a CANFD message payload overflow condition occurs.

Do not write to this bit when the CANFD module is in `GL_SLEEP` mode.

Note: This bit is not available in the classical CAN function

### TSRST bit (Timestamp Reset)

When the `TSRST` bit is 1, the Global Timestamp Register is reset to 0x0000.

Do not write to this bit when the CANFD module is in `GL_SLEEP` or `GL_RESET` mode.

Read value is always 0.

This bit is cleared automatically by the CANFD module logic.



### 32.2.13 CFDGSTS : Global Status Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x001C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	GRAM INIT	GSLP STS	GHLT STS	GRST STS	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Bit	Symbol	Function	R/W
0	GRSTSTS	Global Reset Status 0: Not in Reset mode 1: In Reset mode	R
1	GHLTSTS	Global Halt Status 0: Not in Halt mode 1: In Halt mode	R
2	GSLPSTS	Global Sleep Status 0: Not in Sleep mode 1: In Sleep mode	R
3	GRAMINIT	Global RAM Initialization 0: RAM initialization is complete 1: RAM initialization is ongoing	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The Global Status Register indicates the global status of the CANFD module.

#### GRSTSTS bit (Global Reset Status)

The GRSTSTS bit indicates the status of Global CANFD module Reset mode.

This bit is set automatically when the CANFD module enters GL\_RESET mode. When the mode changes from GL\_RESET mode to GL\_SLEEP mode, this bit remains set.

This bit is cleared automatically when the CANFD module exits the GL\_RESET mode.

#### GHLTSTS bit (Global Halt Status)

The GHLTSTS bit indicates the status of Global CANFD module Halt mode.

This bit is set automatically when the CANFD module enters GL\_HALT mode.

This bit is cleared automatically when the CANFD module exits the GL\_HALT mode.

#### GSLPSTS bit (Global Sleep Status)

The GSLPSTS bit indicates the status of Global CANFD module Sleep mode.

This bit is set automatically when the CANFD module enters GL\_SLEEP mode.

This bit is cleared automatically when the CANFD module exits the GL\_SLEEP mode.

#### GRAMINIT bit (Global RAM Initialization)

The GRAMINIT bit indicates the status of Global CANFD module RAM initialization.

This bit is set automatically when the CANFD module enters GL\_SLEEP mode after a hardware reset.

This bit is cleared automatically when the CANFD module completed RAM initialization.

This bit is cleared when the test\_mode input port is set to 1.

### 32.2.14 CFDGERFL : Global Error Flag Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EEF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CMPO F	THLE S	MES	DEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DEF	DLC Error Flag 0: DLC error not detected 1: DLC error detected	R/W
1	MES	Message Lost Error Status 0: Message lost error not detected 1: Message lost error detected	R
2	THLES	TX History List Entry Lost Error Status 0: TX history list entry lost error not detected 1: TX history list entry lost error detected	R
3	CMPOF <sup>*1</sup>	CANFD Message Payload Overflow Flag 0: CANFD message payload overflow not detected 1: CANFD message payload overflow detected	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
16	EEF0	ECC Error Flag 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is not available in the classical CAN function.

The Global Error Flag register indicates the detection of global errors.

#### DEF bit (DLC Error Flag)

The DEF bit indicates the error status of the DLC.

Do not write to this bit when the CANFD module is in GL\_SLEEP or GL\_RESET mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when a DLC error is detected in a received frame.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set

The bit is cleared by writing 0 to it.

This bit is cleared automatically in GL\_RESET mode.

#### MES bit (Message Lost Error Status)

The MES bit indicates status of the message lost error.

This bit is set automatically when a FIFO message lost error is detected.

This bit is cleared automatically when:

- All FIFO message lost flags are cleared
- The CANFD module is in GL\_RESET mode.

**THLES bit (TX History List Entry Lost Error Status)**

The THLES bit indicates status of the TX history list entry lost error.

This bit is set automatically when a TX history list entry lost error is detected.

This bit is cleared automatically when:

- All TX history list entry lost flags are cleared
- The CANFD module is in GL\_RESET mode.

**CMPOF bit (CANFD Message Payload Overflow Flag)**

The CMPOF bit is set automatically when a CANFD message payload overflow is detected on at least one channel.

Do not write to this bit when the CANFD module is in GL\_SLEEP or GL\_RESET mode.

This bit is cleared by writing 0 to it. Writing 1 to this bit has no effect.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared automatically in GL\_RESET mode.

Note: This bit is not available in the classical CAN function

**EEF0 bit (ECC Error Flag)**

The EEF0 bit specifies whether an ECC error has occurred.

Do not write to this bit when the CANFD module is in GL\_SLEEP or GL\_RESET mode. Writing 1 to this bit has no effect.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The bit is cleared by writing 0 to it. This bit is cleared automatically in GL\_RESET mode.

**32.2.15 CFDGTINTSTS : Global TX Interrupt Status Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00A4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	THIF0	CFTIF0	TQIF0	TAIO	TSIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSIF0	TX Successful Interrupt Flag 0: Channel n TX Successful Interrupt flag not set 1: Channel n TX Successful Interrupt flag set	R
1	TAIO	TX Abort Interrupt Flag 0: Channel n TX Abort Interrupt flag not set 1: Channel n TX Abort Interrupt flag set	R

Bit	Symbol	Function	R/W
2	TQIF0	TX Queue Interrupt Flag 0: Channel n TX Queue Interrupt flag not set 1: Channel n TX Queue Interrupt flag set	R
3	CFTIF0	COM FIFO TX Mode Interrupt Flag 0: Channel n COM FIFO TX Mode Interrupt flag not set 1: Channel n COM FIFO TX Mode Interrupt flag set	R
4	THIF0	TX History List Interrupt 0: Channel n TX History List Interrupt flag not set 1: Channel n TX History List Interrupt flag set	R
31:5	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The Global TX Interrupt Status register indicates the detection of transmit specific interrupts.

### TSIF0 bit (TX Successful Interrupt Flag)

The TSIF0 bit is set to 1 when the TX Successful Interrupt flag of the related channel is set (when the interrupt is enabled). This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL\_RESET or CH\_RESET mode.

### TAI0 bit (TX Abort Interrupt Flag)

The TAI0 bit is set to 1 when the TX Abort Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL\_RESET or CH\_RESET mode.

### TQIF0 bit (TX Queue Interrupt Flag)

The TQIF0 bit is set to 1 when the TX Queue Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX Queue Interrupt flag is cleared (when the interrupt is enable disabled)
- When in GL\_RESET or CH\_RESET mode.

### CFTIF0 bit (COM FIFO TX Mode Interrupt Flag)

The CFTIF0 bit is set to 1 when the related COM TX FIFO Mode Interrupt flag (CFDCFSTS.CFTXIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related COM TX FIFO Mode Interrupt flag (CFDCFSTS.CFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL\_RESET or CH\_RESET mode.

### THIF0 bit (TX History List Interrupt)

The THIF0 bit is set to 1 when the related TX History List Interrupt flag (CFDTHLSTS.THLIF) is set (when the interrupt is enabled).

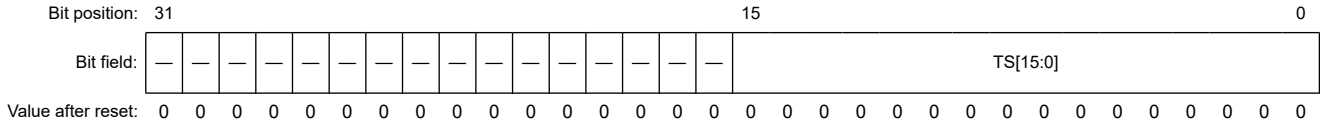
This bit is cleared automatically:

- When the related TX History List Interrupt flag (CFDTHLSTS.THLIF) is cleared (when the interrupt enable is disabled)
- When in GL\_RESET or CH\_RESET mode.

### 32.2.16 CFDGTSC : Global Timestamp Counter Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0024



Bit	Symbol	Function	R/W
15:0	TS[15:0]	Timestamp value	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The Global Timestamp Counter register stores the timestamp based on the selected configuration.

#### TS[15:0] bits (Timestamp value)

The Timestamp value is stored in the Global Timestamp Counter register based on the configuration of TSSS, TSBTCS and TSP. The accuracy of the timestamp counter cannot be guaranteed when transitioning to halt state.

The Timestamp value is stored in this register based on the configuration of TSSS, TSBTCS and TSP.

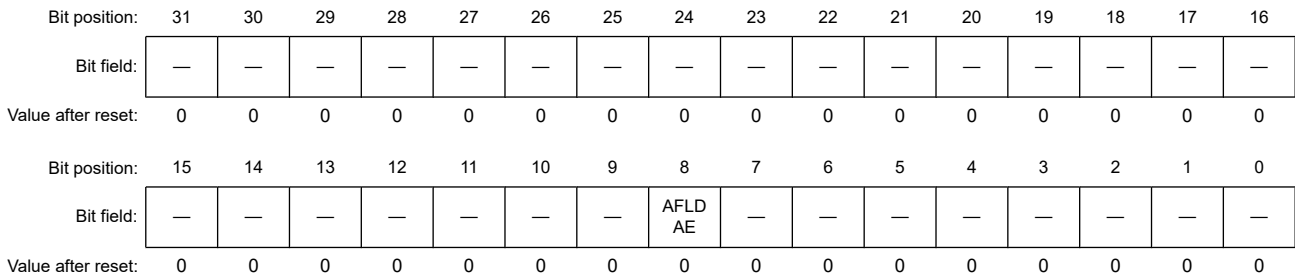
Do not write to bits TS[15:0] when the CANFD module is in GL\_RESET or GL\_SLEEP mode.

The TS[15:0] bits are cleared automatically in GL\_RESET mode.

### 32.2.17 CFDGAFLECTR : Global Acceptance Filter List Entry Control Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0028



Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	AFLDAE	Acceptance Filter List Data Access Enable 0: Acceptance Filter List data access disabled 1: Acceptance Filter List data access enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Entry Control Register is used to select the Global Acceptance Filter List page for reading or writing entries into the Global Acceptance Filter List.

#### AFLDAE bit (Acceptance Filter List Data Access Enable)

The AFLDAE bit prevents write access to the Acceptance Filter List when cleared after configuration of the Acceptance Filter List.

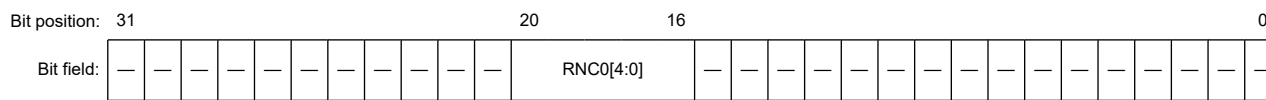
Data can be read from the Acceptance Filter List independent of the status of this bit.

Do not write to this bit when the CANFD module is in GL\_SLEEP mode. Set this bit to enable write access for the Acceptance Filter List.

### 32.2.18 CFDGAFLECFG : Global Acceptance Filter List Configuration Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x002C



Value after reset: 0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
20:16	RNC0[4:0]	Rule Number Number of rules dedicated	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Configuration Register is used to define the number of rules for entries in the Acceptance Filter List.

The total number of available entries in the Acceptance Filter List is 16.

#### RNC0[4:0] bits (Rule Number)

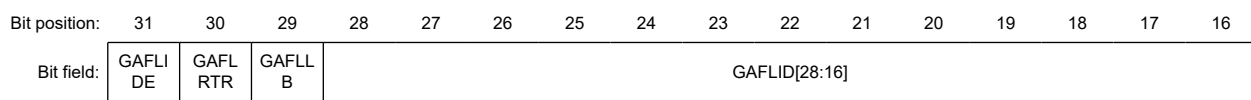
The RNC0[4:0] bits define the number of rules in the Acceptance Filter List.

Only write to these bits when the CANFD module is in GL\_RESET mode. These bits can set to 5 bits for 16 rules.

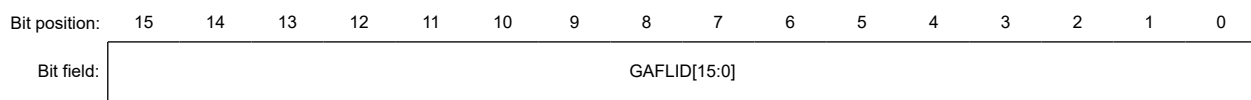
### 32.2.19 CFDGAFLEIDr : Global Acceptance Filter List ID Registers (r = 1 to 16)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0120 + 0x0010 × (r - 1)



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
28:0	GAFLID[28:0]	Global Acceptance Filter List Entry ID Field ID part of the Global Acceptance Filter List entry	R/W
29	GAFLLB	Global Acceptance Filter List Entry Loopback Configuration 0: Global Acceptance Filter List entry ID for acceptance filtering with attribute RX 1: Global Acceptance Filter List entry ID for acceptance filtering with attribute TX	R/W
30	GAFLRTR	Global Acceptance Filter List Entry RTR Field 0: Data frame 1: Remote frame	R/W

Bit	Symbol	Function	R/W
31	GAFLIDE	Global Acceptance Filter List Entry IDE Field 0: Standard identifier of rule entry ID is valid for acceptance filtering 1: Extended identifier of rule entry ID is valid for acceptance filtering	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List ID Registers are used to configure the ID field for the rules of entries in the Global Acceptance Filter List.

### GAFLID[28:0] bits (Global Acceptance Filter List Entry ID Field)

The GAFLID[28:0] bits represent the CAN identifier (ID) field of each entry in the Global Acceptance Filter List.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

### GAFLLB bit (Global Acceptance Filter List Entry Loopback Configuration)

The GAFLLB bit selects whether entry in the Global Acceptance Filter List gets the attribute RX or TX.

This attribute determines the validity of the entry in Mirror mode, Loopback test mode, and during standard (non-loopback) reception. See [section 32.5.5. Loopback Modes](#) for detailed description of the validity of the Global Acceptance Filter List entry depending on transmitter/receiver case, the type of loopback mode, and RX/TX attribute.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

### GAFLRTR bit (Global Acceptance Filter List Entry RTR Field)

The GAFLRTR bit allows the configuration of the specified frame format (data frame or remote frame) for each entry of the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the RTR bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

### GAFLIDE bit (Global Acceptance Filter List Entry IDE Field)

The GAFLIDE bit allows the configuration of the ID format (standard ID or extended ID) for each entry in the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the IDE bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

## 32.2.20 CFDGAFLMr : Global Acceptance Filter List Mask Registers (r = 1 to 16)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0124 + 0x0010 × (r - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLI DEM	GAFL RTRM	GAFLI FL1	GAFLIDM[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLIDM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	GAFLIDM[28:0]	Global Acceptance Filter List ID Mask Field Global Acceptance Filter List Mask field bits for ID field	R/W
29	GAFLIFL1	Global Acceptance Filter List Information Label 1 Global Acceptance Filter List information label bit 1	R/W
30	GAFLRTRM	Global Acceptance Filter List Entry RTR Mask 0: RTR bit is not used for ID matching 1: RTR bit is used for ID matching	R/W
31	GAFLIDEM	Global Acceptance Filter List IDE Mask 0: IDE bit is not used for ID matching 1: IDE bit is used for ID matching	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Mask Registers are used to configure the Mask field of each rule for entries in the Global Acceptance Filter List.

### GAFLIDM[28:0] bits (Global Acceptance Filter List ID Mask Field)

GAFLIDM[28:0] bits are the filter mask bits for the related bits in the CAN Identifier field of each Global Acceptance Filter List entry.

0	Corresponding STD-ID/EXT-ID bit is not used for ID matching
1	Corresponding STD-ID/EXT-ID bit is used for ID matching

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

### GAFLIFL1 bit (Global Acceptance Filter List Information Label 1)

The GAFLIFL1 bit allows the configuration of a 2-bit information label to be attached to a received message accepted by the associated entry in the Global Acceptance Filter List. This bit is a MSB bit of an information label.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

This bit is stored in the Information Label Field [1] (CFDRMFDSTSb.RMIFL [1], CFDRFFDSTSb.RFIFL [1], CFDCFFDCSTS.CFIFL [1]) of the storage location of an incoming message.

### GAFLRTRM bit (Global Acceptance Filter List Entry RTR Mask)

The GAFLRTRM bit allows the configuration of the RTR mask bit for each entry in the Global Acceptance Filter List.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

### GAFLIDEM bit (Global Acceptance Filter List IDE Mask)

The GAFLIDEM bit allows the configuration of the IDE mask bit for each entry in the Global Acceptance Filter List.

When the IDE mask bit is 0, the ID comparison depends on the received IDE bit.

If the received IDE bit is 0, the STD-ID comparison takes place.

If the received IDE bit is 1, the EXT-ID comparison takes place.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

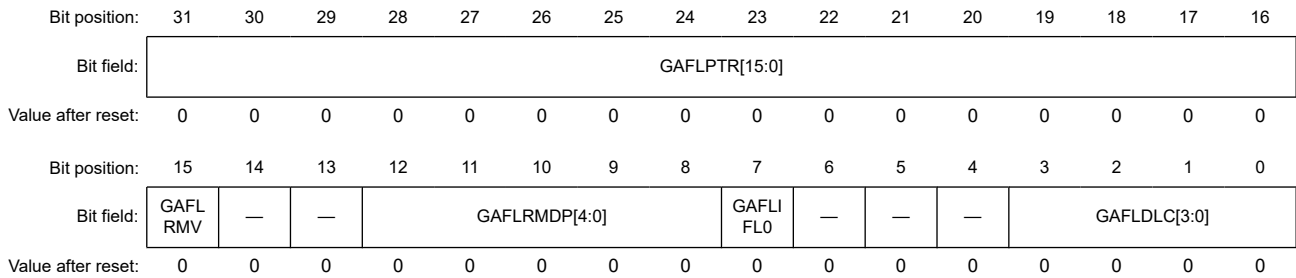
Only write to this bit when the related CANFD channel is in CH\_RESET or CH\_HALT mode.



### 32.2.21 CFDGAFLP0r : Global Acceptance Filter List Pointer 0 Registers (r = 1 to 16)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0128 + 0x0010 × (r - 1)



Bit	Symbol	Function	R/W
3:0	GAFLDLC[3:0]	Global Acceptance Filter List DLC Field Minimum number of data bytes in a data frame required for acceptance	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	GAFLIFL0	Global Acceptance Filter List Information Label 0	R/W
12:8	GAFLRMDP[4:0]	Global Acceptance Filter List RX Message Buffer Direction Pointer RX message buffer number for storage of received messages	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	GAFLRMV	Global Acceptance Filter List RX Message Buffer Valid 0: Single message buffer direction pointer is invalid 1: Single message buffer direction pointer is valid	R/W
31:16	GAFLPTR[15:0]	Global Acceptance Filter List Pointer	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Pointer 0 Registers are used to configure the data length code (DLC), software pointer, single message buffer select, and message buffer direction pointer for each rule entry in the Global Acceptance Filter List.

#### GAFLDLC[3:0] bits (Global Acceptance Filter List DLC Field)

The GAFLDLC[3:0] bits allow the configuration of a minimum data length code (DLC) value for a message to be accepted by the associated entry in the Global Acceptance Filter List (automatic DLC filter function).

DLC filter process is only passed if the DLC value of the message accepted by an entry in the Global Acceptance Filter List is equal to or higher than the DLC value configured for this associated Global Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0.

Table 32.3 shows DLC value that can be configured.

**Table 32.3 Configuration of DLC value (1 of 2)**

Format	DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
CAN and CANFD	0	0	0	0	DLC of received message = 0 or more (DLC filter check is disabled)
CAN and CANFD	0	0	0	1	DLC of received message = 1 or more
CAN and CANFD	0	0	1	0	DLC of received message = 2 or more
CAN and CANFD	0	0	1	1	DLC of received message = 3 or more
CAN and CANFD	0	1	0	0	DLC of received message = 4 or more
CAN and CANFD	0	1	0	1	DLC of received message = 5 or more

**Table 32.3 Configuration of DLC value (2 of 2)**

Format	DLC[3]	DLC[2]	DLC[1]	DLC[0]	Description
CAN and CANFD	0	1	1	0	DLC of received message = 6 or more
CAN and CANFD	0	1	1	1	DLC of received message = 7 or more
CAN	1	x	x	x	DLC of received message = 8 or more
CANFD	1	0	0	0	DLC of received message = 8 or more <sup>*1</sup>
CANFD	1	0	0	1	DLC of received message = 12 or more <sup>*1</sup>
CANFD	1	0	1	0	DLC of received message = 16 or more <sup>*1</sup>
CANFD	1	0	1	1	DLC of received message = 20 or more <sup>*1</sup>
CANFD	1	1	0	0	DLC of received message = 24 or more <sup>*1</sup>
CANFD	1	1	0	1	DLC of received message = 32 or more <sup>*1</sup>
CANFD	1	1	1	0	DLC of received message = 48 or more <sup>*1</sup>
CANFD	1	1	1	1	DLC of received message = 64 <sup>*1</sup>

Note 1. This setting is not available in the classical CAN function.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

#### **GAFLIFL0 bit (Global Acceptance Filter List Information Label 0)**

The GAFLIFL0 bit allows the configuration of a 2-bit information label that can be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a LSB bit of an information label.

You cannot write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

This bit is stored in Information Label Field[0] (CFDRMFDSTSb.RMIFL[0], CFDRFFDSTSb.RFIFL[0], CFDCFFDCSTS.CFIFL[0]) of the storage location of an incoming message.

#### **GAFLRMDP[4:0] bits (Global Acceptance Filter List RX Message Buffer Direction Pointer)**

The GAFLRMDP[4:0] bits allow the configuration of a single reception message buffer as the destination target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry. The value entered is the single destination message buffer number.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

CFDRMNB.NRXMB[4:0] is the value entered in the RX Message Buffer Number Register to configure the number of RX message buffers. The value to be entered in CFDGAFLP0r.GAFLRMDP[4:0] bits should only be between 0x00 and CFDMNB.NMB[4:0] to 1 less.

If CFDRMNB.NRXMB[4:0] = 0x00, the GAFLRMV bit should be configured as 0.

#### **GAFLRMV bit (Global Acceptance Filter List RX Message Buffer Valid)**

The GAFLRMV bit allows the enabling or disabling of a single reception message buffer as the target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

#### **GAFLPTR[15:0] bits (Global Acceptance Filter List Pointer)**

The GAFLPTR[15:0] bits allow the configuration of a 16-bit pointer to be attached to a received message accepted by the related Global Acceptance Filter List entry. The pointer is added during message storage in the Message Buffer area and can be used by the application as a support function. The pointer information can be used for example, to support PDU Identifier allocation for the received message in AUTOSAR systems.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

### 32.2.22 CFDGAF1P1r : Global Acceptance Filter List Pointer 1 Registers (r = 1 to 16)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x012C + 0x0010 × (r - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	GAFL FDP8	—	—	—	—	—	—	GAFL FDP1	GAFL FDP0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GAFLFDP0	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage 0: Disable RX FIFO 0 as target for reception 1: Enable RX FIFO 0 as target for reception	R/W
1	GAFLFDP1	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage 0: Disable RX FIFO 1 as target for reception 1: Enable RX FIFO 1 as target for reception	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	GAFLFDP8	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage 0: Disable Common FIFO as target for reception 1: Enable Common FIFO as target for reception	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Acceptance Filter List Pointer 1 registers are used to configure the FIFO direction pointer fields in each Rule Entry of the Global Acceptance Filter List.

#### GAFLFDP8, GAFLFDP1, GAFLFDP0 bits (Global Acceptance Filter List FIFO Direction Pointer)

These bits allow the configuration of FIFO Buffers as the target for a received message passing the acceptance check of the related Global Acceptance Filter List entry. Each bit of the GAFLFDP8, GAFLFDP1, GAFLFDP0 is configuring a dedicated FIFO.

Users cannot write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

For storage in Common FIFO, target for reception can only be those Common FIFO Buffers that are configured as RX FIFO.

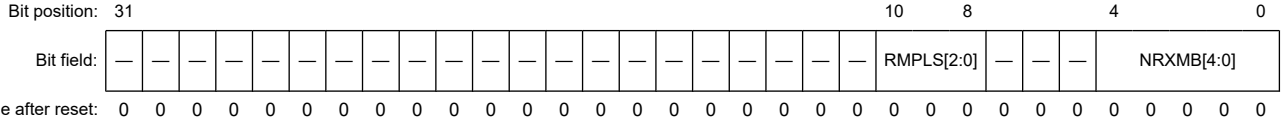
Only write to these bits when the related CANFD channel is in CH\_RESET or CH\_HALT mode.

Users should only configure up to 2 destination FIFO Buffers or 1 destination FIFO Buffers plus one RX Message Buffer.

### 32.2.23 CFDRMNB : RX Message Buffer Number Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0030



Bit	Symbol	Function	R/W
4:0	NRXMB[4:0]	Number of RX Message Buffers	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RMPLS[2:0]	Reception Message Buffer Payload Data Size 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The RX Message Buffer Number register is used to configure the total number of RX message buffers allocated to channels.

#### NRXMB[4:0] bits (Number of RX Message Buffers)

The NRXMB[4:0] bits are used to configure the number of RX message buffers.

Only write to these bits when the CANFD module is in GL\_RESET mode.

Enter only values between 0 and 16 inclusive, with 0x00 indicating that no RX message buffer is allocated.

#### RMPLS[2:0] bits (Reception Message Buffer Payload Data Size)

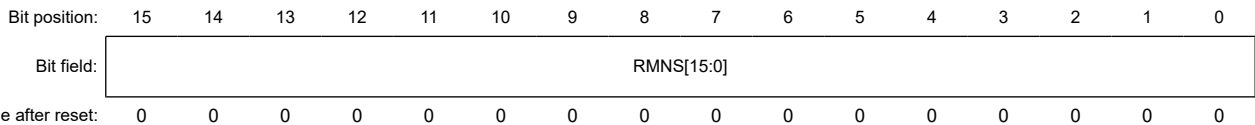
The RMPLS[2:0] bits are used to configure the message buffer payload data size.

Only write to these bits when the CANFD module is in GL\_RESET mode.

### 32.2.24 CFDRMND : RX Message Buffer New Data Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0034



Bit	Symbol	Function	R/W
15:0	RMNS[15:0]	RX Message Buffer New Data Status 0: New data not stored in corresponding RX message buffer 1: New data stored in corresponding RX message buffer	R/W

Note: S-TYPE-3, P-TYPE-3

The RX Message Buffer New Data Status Register specifies the new data storage status of the RX message buffers.

**RMNS[15:0] bits (RX Message Buffer New Data Status)**

The RMNS[15:0] bits indicate the status of new data for the corresponding RX message buffer. RMNS bit [0] corresponds to RX message buffer [0] and so on.

The bit position of CFDRMND corresponds to the buffer number of RXMB.

Do not write to these bits when the CANFD module is in GL\_RESET or GL\_SLEEP mode. Writing 1 has no effect.

These bits cannot be cleared when message storage in the corresponding RX message buffer is in progress.

Do not use the bit clear instruction to clear these bits. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

These bits are set automatically when storage of new messages are in the corresponding RX message buffer. These bits are cleared by writing 0. These bits are cleared automatically when the CANFD module is in GL\_RESET mode.

When CFDRMNB.RMPLS = 000b (maximum 8 bytes payload), the duration of message storage is 6 PCLKA cycles.

When CFDRMNB.RMPLS > 000b, the duration of message storage is 6 PCLKA cycles + 1 for each 4 bytes (maximum of 20 PCLKA cycles for 64 bytes).

Note: This feature is not available in the classical CAN function.

**32.2.25 CFDRFCCa : RX FIFO Configuration/Control Registers a (a = 0 to 1)**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x003C + 0x04 × a

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	RFPLS[2:0]			—	—	RFIE	RFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFE	RX FIFO Enable 0: FIFO disabled 1: FIFO enabled	R/W
1	RFIE	RX FIFO Interrupt Enable 0: FIFO interrupt generation disabled 1: FIFO interrupt generation enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
6:4	RFPLS[2:0] <sup>*1</sup>	Rx FIFO Payload Data Size Configuration 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10:8	RFDC[2:0]	RX FIFO Depth Configuration 0 0 0: FIFO Depth = 0 message 0 0 1: FIFO Depth = 4 messages 0 1 0: FIFO Depth = 8 messages 0 1 1: FIFO Depth = 16 messages 1 0 0: FIFO Depth = 32 messages 1 0 1: FIFO Depth = 48 messages 1 1 0: Reserved 1 1 1: Reserved	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	RFIM	RX FIFO Interrupt Mode 0: Interrupt generated when RX FIFO counter reaches RFIGCV value from values smaller than RFIGCV 1: Interrupt generated at the end of every received message storage	R/W
15:13	RFIGCV[2:0]	RX FIFO Interrupt Generation Counter Value 0 0 0: Interrupt generated when FIFO is 1/8th full 0 0 1: Interrupt generated when FIFO is 1/4th full 0 1 0: Interrupt generated when FIFO is 3/8th full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8th full 1 0 1: Interrupt generated when FIFO is 3/4th full 1 1 0: Interrupt generated when FIFO is 7/8th full 1 1 1: Interrupt generated when FIFO is full	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. These bits are not available in the classical CAN function.

The RX FIFO Configuration/Control Registers are used to configure and control the two RX FIFOs.

#### RFE bit (RX FIFO Enable)

The RFE bit enables the FIFO. When this bit is set to 0, the RX FIFO is cleared to empty.

Only write to this bit when the CANFD module is in GL\_HALT or GL\_OPERATION mode.

This bit can only be set if the configured FIFO depth is greater than 0x000 (CFDRFCCa.RFDC > 0x000) and less than 0x110.

Set the RFE bit with a separate write access to the CFDRFCCa register, after all the other bits in the CFDRFCCa register are set.

This bit is cleared automatically when the CANFD module is in GL\_RESET mode.

#### RFIE bit (RX FIFO Interrupt Enable)

The RFIE bit enables generation of the FIFO interrupt.

Do not write to this bit when the CANFD module is in GL\_SLEEP mode.

#### RFPLS[2:0] bits (Rx FIFO Payload Data Size Configuration)

The RFPLS[2:0] bits define the message data payload allocation in the RAM.

This is the maximum number of bytes which can be received by this FIFO.

Only write to these bits when the CANFD module is in GL\_RESET mode.

Note: These bits are not available in the classical CAN function.

#### RFDC[2:0] bits (RX FIFO Depth Configuration)

The RFDC[2:0] bits select the depth of the FIFO in terms of the number of messages. If the FIFO depth is configured to 0 messages, the FIFO cannot be used.

Only write to these bits when the CANFD module is in GL\_RESET mode.

**RFIM bit (RX FIFO Interrupt Mode)**

The RFIM bit selects the interrupt generation condition for the FIFO.

Do not write to this bit when the CANFD module is in GL\_SLEEP mode.

Only write to this bit when the CANFD module is in GL\_RESET mode.

**RFIGCV[2:0] bits (RX FIFO Interrupt Generation Counter Value)**

The RFIGCV[2:0] bits select the counter value of the FIFO for generation of FIFO interrupts. These values represent fractions of the FIFO depth for which an interrupt is generated.

Do not write to these bits when the CANFD module is in GL\_SLEEP mode.

The setting of the RFIGCV[2:0] bits should be synchronized with the RFDC[2:0] bits.

Only write to these bits when the CANFD module is in GL\_RESET mode.

**32.2.26 CFDRFSTSa : RX FIFO Status Registers a (a = 0 to 1)**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0044 + 0x04 × a

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RFMC[5:0]					—	—	—	—	RFIF	RFMLT	RFLL	RFEMP	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	RFEMP	RX FIFO Empty 0: FIFO not empty 1: FIFO empty	R
1	RFLL	RX FIFO Full 0: FIFO not full 1: FIFO full	R
2	RFMLT	RX FIFO Message Lost 0: No message lost in FIFO 1: FIFO message lost	R/W
3	RFIF	RX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied 1: FIFO interrupt condition satisfied	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
13:8	RFMC[5:0]	RX FIFO Message Count Number of messages stored in FIFO	R
31:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The RX FIFO Status Registers show the status of messages stored in the corresponding FIFO buffers.

**RFEMP bit (RX FIFO Empty)**

The RFEMP bit is set automatically when:

- The RFMC bit is 0
- RX FIFO is disabled by setting the CFDRFCCa.RFE bit to 0
- The CANFD module is in GL\_RESET mode.

The RFEMP bit is cleared automatically when the first message is stored in the RX FIFO buffer.

#### RFFLL bit (RX FIFO Full)

The RFFLL bit is set automatically when the number of CAN messages stored in the FIFO buffer matches the configured FIFO depth.

The RFFLL is cleared automatically when:

- The number of CAN messages stored in the FIFO buffer is less than the configured FIFO depth
- RX FIFO is disabled by setting the CFDRFCCa.RFE bit to 0
- The CANFD module is in GL\_RESET mode.

#### RFMLT bit (RX FIFO Message Lost)

Only write to the RFMLT bit when CANFD module is in GL\_HALT or GL\_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically whenever a message is lost due to attempted storage when the FIFO buffer is already full. If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL\_RESET mode.

#### RFIF bit (RX FIFO Interrupt Flag)

The RFIF bit is set automatically when the configured interrupt condition is satisfied. This bit is not automatically cleared when the RX FIFO buffer is disabled.

Only write to this bit when the CANFD module is in GL\_HALT or GL\_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

The bit is cleared by writing 0 to it. The bit is also cleared when CANFD module is in GL\_RESET mode.

#### RFMC[5:0] bits (RX FIFO Message Count)

The RFMC[5:0] bits indicate the number of CAN messages stored in the RX FIFO buffer that can be read by the CPU.

These bits are cleared automatically when the FIFO is disabled and when the CANFD module is in GL\_RESET mode.

### 32.2.27 CFDRFPCTRa : RX FIFO Pointer Control Registers a (a = 0 to 1)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x004C + 0x04 × a



Bit	Symbol	Function	R/W
7:0	RFPC[7:0]	RX FIFO Pointer Control Increments read pointer of the corresponding RX FIFO buffers	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The RX FIFO Pointer Control Registers can be used to increment the read pointer of the corresponding RX FIFO buffers.



**RFPC bits (RX FIFO Pointer Control)**

When the value 0xFF is written to the RFPC bits, the pointer of the corresponding RX FIFO buffer is moved to the next FIFO entry. Only write 0xFF to these registers when the corresponding RX FIFO buffer is enabled and not empty.

The read value from these bits is always 0x00.

Only write to these bits when the CANFD module is in GL\_HALT or GL\_OPERATION mode.

Do not write to the RX FIFO Pointer Control registers when DMA is enabled.

**32.2.28 CFDCFCC : Common FIFO Configuration/Control Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0054

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFITT[7:0]							CFDC[2:0]			—	—	—	CFTML[1:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CFIGCV[2:0]		CFIM	CFITR	CFITSS	—	CFM	—	CFPLS[2:0]			—	CFTXIE	CFRXIE	CFE	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFE	Common FIFO Enable 0: FIFO disabled 1: FIFO enabled	R/W
1	CFRXIE	Common FIFO RX Interrupt Enable 0: FIFO interrupt generation disabled for Frame RX 1: FIFO interrupt generation enabled for Frame RX	R/W
2	CFTXIE	Common FIFO TX Interrupt Enable 0: FIFO interrupt generation disabled for Frame TX 1: FIFO interrupt generation enabled for Frame TX	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CFPLS[2:0]*1	Common FIFO Payload Data Size Configuration 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	CFM	Common FIFO Mode 0: RX FIFO mode 1: TX FIFO mode	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	CFITSS	Common FIFO Interval Timer Source Select 0: Reference clock (× 1 / × 10 period) 1: Bit time clock of related channel (FIFO is linked to fixed channel)	R/W
11	CFITR	Common FIFO Interval Timer Resolution 0: Reference clock period × 1 1: Reference clock period × 10	R/W

Bit	Symbol	Function	R/W
12	CFIM	Common FIFO Interrupt Mode 0: RX FIFO mode: RX interrupt generated when Common FIFO counter reaches CFGICV value from a lower value TX FIFO mode: TX interrupt generated when Common FIFO transmits the last message successfully 1: RX FIFO mode: RX interrupt generated at the end of every received message storage TX FIFO mode: interrupt generated for every successfully transmitted message	R/W
15:13	CFGICV[2:0]	Common FIFO Interrupt Generation Counter Value 0 0 0: Interrupt generated when FIFO is 1/8th full 0 0 1: Interrupt generated when FIFO is 1/4th full 0 1 0: Interrupt generated when FIFO is 3/8th full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8th full 1 0 1: Interrupt generated when FIFO is 3/4th full 1 1 0: Interrupt generated when FIFO is 7/8th full 1 1 1: Interrupt generated when FIFO is full	R/W
17:16	CFTML[1:0]	Common FIFO TX Message Buffer Link Transmission scan link position of the corresponding channel	R/W
20:18	—	These bits are read as 0. The write value should be 0.	R/W
23:21	CFDC[2:0]	Common FIFO Depth Configuration 0 0 0: FIFO Depth = 0 message 0 0 1: FIFO Depth = 4 messages 0 1 0: FIFO Depth = 8 messages 0 1 1: FIFO Depth = 16 messages 1 0 0: FIFO Depth = 32 messages 1 0 1: FIFO Depth = 48 messages 1 1 0: FIFO Depth = Reserved 1 1 1: FIFO Depth = Reserved	R/W
31:24	CFITT[7:0]	Common FIFO Interval Transmission Time Delay the start of transmission from the FIFO if configured in TX mode, delay is a multiple of basic Interval Timer Clock Source unit	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. These bits are not available in the classical CAN function.

### CFE bit (Common FIFO Enable)

The CFE bit enables the FIFO when set. FIFO is disabled when this bit is cleared.

This bit can also be used, by clearing it, to abort transmission from Common FIFO when configured in TX mode, or to stop reception into the Common FIFO in RX mode.

Only write to this bit when the CANFD module is in GL\_HALT or GL\_OPERATION mode and the related CANFD channel is not in CH\_RESET mode for FIFOs configured as TX FIFO.

This bit can only be set if the configured FIFO depth is greater than 0x000 (CFDCFCC.CFDC > 0x000) and less than 0x110 (0x110 > CFDCFCC.CFDC > 0x000).

Set the CFE bit with a separate write access to the CFDCFCC register, after all the other bits in this register are set.

This bit is cleared automatically when the CANFD module is in GL\_RESET mode.

This bit is also cleared automatically when the related channel is in CH\_RESET mode if the FIFO is configured in TX mode.

### CFRXIE bit (Common FIFO RX Interrupt Enable)

The CFRXIE bit enables generation of FIFO interrupts when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CANFD module is in GL\_SLEEP mode.

### CFTXIE bit (Common FIFO TX Interrupt Enable)

The CFTXIE bit enables generation of common FIFO interrupts when the interrupt flag is set after transmission of a frame from the corresponding FIFO buffer.

Do not write to this bit when the CANFD module is in GL\_SLEEP mode.

### **CFPLS[2:0] bits (Common FIFO Payload Data Size Configuration)**

The CFPLS[2:0] bits define the message data payload allocation in the RAM. This is the maximum number of bytes which can be received or transmitted by the FIFO buffer.

For details, see [section 32.6. FIFO Buffers and Normal Message Buffer Configuration](#).

Only write to this bit when the CANFD module is in GL\_RESET mode.

Note: These bits are not available in the classical CAN function.

### **CFM bit (Common FIFO Mode)**

The CFM bit selects the mode of the FIFO. When a hardware reset is applied, all the Common FIFO buffers are configured in RX FIFO mode.

Do not write to these bits in GL\_OPERATION or GL\_SLEEP mode.

Only write to these bits when the CANFD module is in GL\_RESET mode.

### **CFITSS bit (Common FIFO Interval Timer Source Select)**

The CFITSS bit selects the basic clock source for the Interval Transmission Timer.

Do not write to this bit when the CANFD module is in GL\_SLEEP mode. In addition, do not write to this bit when the CFE bit is set to 1.

Do not write 1 to this bit when CANFD communication is used.\*1

Note: The bit time clock can vary depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

### **CFITR bit (Common FIFO Interval Timer Resolution)**

The CFITR bit selects the resolution of the reference clock for the Interval Transmission Timer (peripheral clock is the source for the reference clock).

Do not write to this bit when the CANFD module is in GL\_SLEEP mode. Also, do not write to this bit when the CFE bit is set to 1.

### **CFIM bit (Common FIFO Interrupt Mode)**

The CFIM bit selects the interrupt generation condition for the FIFO buffer.

Do not write to this bit in GL\_SLEEP mode.

Only write to this bit when the CANFD module is in GL\_RESET mode.

### **CFIGCV[2:0] bits (Common FIFO Interrupt Generation Counter Value)**

The CFIGCV[2:0] bits select the message counter value for the generation of FIFO interrupts. These values represent fractions of the FIFO depth at which the interrupt is to be generated.

Do not write to these bits when the CANFD module is in GL\_SLEEP mode.

The setting of these bits should be synchronized with the CFDC[2:0] bits.

Only write to these bits when the CANFD module is in GL\_RESET mode.

### **CFTML[1:0] bits (Common FIFO TX Message Buffer Link)**

The CFTML[1:0] bits select the normal transmit message buffer position where the TX FIFO is linked to, for transmission scanning.

Do not write to these bits in GL\_OPERATION or GL\_SLEEP mode.

Only write to this bit when the CANFD module is in GL\_RESET mode.

### **CFDC[2:0] bits (Common FIFO Depth Configuration)**

The CFDC[2:0] bits select the depth of the common FIFO in terms of the number of messages. If the FIFO depth is configured to 0 message, the FIFO cannot be used.

Only write to these bits when the CANFD module is in GL\_RESET mode.

**CFITT[7:0] bits (Common FIFO Interval Transmission Time)**

The CFITT[7:0] bits select the delay in the start of transmission for all messages transmitted from this FIFO buffer when configured in TX mode. The delay is a multiple of the basic interval timer clock source period (reference clock × 1, reference clock × 10, or bit time clock of the related CAN channel).

Do not write to these bits when the CANFD module is in GL\_SLEEP mode.

Do not write to these bits when the CFE bit is set to 1.

When CFDGCFG.ITRCP[15:0] = 0x0000, set the CFITT[7:0] bits to 0x0000.

**32.2.29 CFDCFSTS : Common FIFO Status Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0058

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CFMC[5:0]					—	—	—	CFTXI F	CFRXI F	CFML T	CFFLL	CFEM P	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	CFEMP	Common FIFO Empty 0: FIFO not empty 1: FIFO empty	R
1	CFFLL	Common FIFO Full 0: FIFO not full 1: FIFO full	R
2	CFMLT	Common FIFO Message Lost 0: Number of message lost in FIFO 1: FIFO message lost	R/W
3	CFRXIF	Common RX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied after frame reception 1: FIFO interrupt condition satisfied after frame reception	R/W
4	CFTXIF	Common TX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied after frame transmission 1: FIFO Interrupt condition satisfied after frame transmission	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
13:8	CFMC[5:0]	Common FIFO Message Count Number of messages stored in FIFO	R
31:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

**CFEMP bit (Common FIFO Empty)**

The CFEMP bit is set automatically when:

- The CPU has read all messages from the FIFO configured in RX mode
- All messages have been transmitted from the FIFO configured in TX mode
- The FIFO is disabled by setting the CFE bit to 0
- The CANFD module is in GL\_RESET mode

- The related CANFD channel is in CH\_RESET when FIFO configured in TX mode.

The CFEMP bit is cleared automatically when:

- The first reception message is stored in the FIFO buffer when configured in RX mode
- The first message to be transmitted is stored in the FIFO buffer when configured in TX mode.

#### **CFLL bit (Common FIFO Full)**

The CFLL bit is set automatically when the number of CAN messages stored in the FIFO matches the configured FIFO depth.

The CFLL bit is cleared automatically when:

- The number of CAN messages stored in the FIFO is less than the configured FIFO depth
- The FIFO is disabled by setting the CFE bit to 0
- The CANFD module is in GL\_RESET mode
- The related CANFD channel is in CH\_RESET mode when FIFO buffer is configured in TX mode.

#### **CFMLT bit (Common FIFO Message Lost)**

The CFMLT bit is set automatically whenever a message is lost due to attempted storage of a new message when FIFO is already full in RX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

Only write to this bit when the CANFD module is in GL\_HALT or GL\_OPERATION mode and the related CANFD channel is not in CH\_RESET mode for FIFO configured as TX FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The CFMLT bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL\_RESET mode
- When the related CANFD channel is in CH\_RESET mode if the FIFO buffer is configured in TX mode.

#### **CFRXIF bit (Common RX FIFO Interrupt Flag)**

The CFRXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CANFD module is in GL\_HALT or GL\_OPERATION mode and the related CANFD channel is not in CH\_RESET mode for FIFO configured as TX FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers when configured in RX mode.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The CFRXIF bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL\_RESET mode

#### **CFTXIF bit (Common TX FIFO Interrupt Flag)**

The CFTXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CANFD module is in GL\_HALT or GL\_OPERATION mode and the related CANFD channel is not in CH\_RESET mode for FIFO buffer configured as TX FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers configured in TX mode.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

The CCTXIF bit is cleared:

- By writing 0 to it
- When the CANFD module is in GL\_RESET mode
- When the related CANFD channel is in CH\_RESET mode if the FIFO buffer is configured in TX mode.

### CFMC[5:0] bits (Common FIFO Message Count)

The CFMC[5:0] bits indicate the following:

- Number of CAN messages stored by the CPU in the FIFO buffer configured in TX mode pending for transmission
- Number of CAN messages stored in the FIFO buffer configured in RX mode by CANFD module to be read by the CPU

The CFMC[5:0] bits are cleared automatically when:

- The FIFO is disabled
- The CANFD module is in GL\_RESET mode
- The related CANFD channel is in CH\_RESET mode if the FIFO buffer is configured in TX mode.

### 32.2.30 CFDCFPCTR : Common FIFO Pointer Control Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

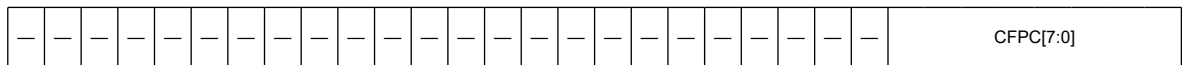
Offset address: 0x005C

Bit position: 31

7

0

Bit field:



Value after reset: 0

Bit	Symbol	Function	R/W
7:0	CFPC[7:0]	Common FIFO Pointer Control Increments read or write pointer of the corresponding Common FIFO buffers depending on the mode configuration.	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The Common FIFO Pointer Control Registers can be used to increment the read or write pointer of the corresponding Common FIFO buffer.

### CFPC[7:0] bits (Common FIFO Pointer Control)

When the value 0xFF is written into the CFPC[7:0] bits, the read pointer of the corresponding Common FIFO buffer (when configured in RX mode), or the write pointer of the corresponding Common FIFO buffer (when configured in TX mode) moves to the next FIFO entry.

The read value from these bits is always 0x00.

Only write to these bits when the CANFD module is in GL\_HALT or GL\_OPERATION mode.

Only write 0xFF to this register when:

- The Common FIFO buffer is enabled and is not empty if configured in RX mode
- The Common FIFO buffer is enabled and is not full if configured in TX mode

Do not write to the Common FIFO Pointer Control registers when DMA is enabled.

### 32.2.31 CFDFESTS : FIFO Empty Status Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFEMP	—	—	—	—	—	—	—	RFXEMP[1:0]
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
1:0	RFXEMP[1:0]	RX FIFO Empty Status 0: Corresponding FIFO not empty 1: Corresponding FIFO empty	R
7:2	—	These bits are read as 0.	R
8	CFEMP	Common FIFO Empty Status 0: Corresponding FIFO not empty 1: Corresponding FIFO empty	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Empty Status register shows status of the empty bits of the FIFO buffers.

#### RFXEMP[1:0] bits (RX FIFO Empty Status)

The RFXEMP[1:0] bits are set when the CANFD module is in GL\_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

#### CFEMP bit (Common FIFO Empty Status)

The CFEMP bits are set when the CANFD module is in GL\_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

### 32.2.32 CFDFFFSTS : FIFO Full Status Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFFLL	—	—	—	—	—	—	—	RFXFLL[1:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RFXFLL[1:0]	RX FIFO Full Status 0: Corresponding FIFO not full 1: Corresponding FIFO full	R
7:2	—	These bits are read as 0.	R
8	CFFLL	Common FIFO Full Status 0: Corresponding FIFO not full 1: Corresponding FIFO full	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Full Status Register shows status of the full bits of the FIFO buffers.

**RFXFLL[1:0] bits (RX FIFO Full Status)**

The RFXFLL[1:0] bits are cleared when CANFD module is in GL\_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

**CFFLL bits (Common FIFO Full Status)**

The CFFLL bits are cleared when the CANFD module is in GL\_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

**32.2.33 CFDFMSTS : FIFO Message Lost Status Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0068

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFMLT	—	—	—	—	—	—	—	RFXMLT[1:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RFXMLT[1:0]	RX FIFO Message Lost Status 0: Corresponding FIFO Message Lost flag not set 1: Corresponding FIFO Message Lost flag set	R
7:2	—	These bits are read as 0.	R
8	CFMLT	Common FIFO Message Lost Status 0: Corresponding FIFO Message Lost flag not set 1: Corresponding FIFO Message Lost flag set	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Message Lost Status Register shows status of the Msg Lost bits of the FIFO buffers.

**RFXMLT[1:0] bits (RX FIFO Message Lost Status)**

The RFXMLT[1:0] bits are cleared when the CANFD module is in GL\_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.



Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

**CFMLT bits (Common FIFO Message Lost Status)**

The CFMLT bits are cleared when the CANFD module is in GL\_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

**32.2.34 CFDRFISTS : RX FIFO Interrupt Flag Status Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x006C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFXIF[1:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RFXIF[1:0]	RX FIFO[x] Interrupt Flag Status 0: Corresponding RX FIFO Interrupt flag not set 1: Corresponding RX FIFO Interrupt flag set	R
31:2	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Interrupt Flag Status Register shows status of the interrupt flag bits of the RX FIFO buffers.

**RFXIF[1:0] bits (RX FIFO[x] Interrupt Flag Status)**

Each bit is set automatically when the corresponding interrupt flag bit is set in the RX FIFO Status Registers.

The RFXIF[1:0] bits are cleared when the CANFD module is in GL\_RESET mode.

Each bit is cleared automatically when the corresponding interrupt flag bit is cleared in the RX FIFO Status Registers.

**32.2.35 CFDCDTCT : DMA Transfer Control Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00C8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFDM AE	—	—	—	—	—	—	—	RFDMAE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFDMAE0	DMA Transfer Enable for RXFIFO 0 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
1	RFDMAE1	DMA Transfer Enable for RXFIFO 1 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	CFDMAE	DMA Transfer Enable for Common FIFO 0 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The DMA Transfer Control Register controls the start and stop of DMA transfer operation.

**RFDMAEe (e = 0 to 1) bit (DMA Transfer Enable for RXFIFO e)**

The RFDMAEe bit cannot be set in GL\_SLEEP or GL\_RESET mode.

This bit is cleared when the CANFD module is in GL\_RESET mode.

**CFDMAE bit (DMA Transfer Enable for Common FIFO)**

The CFDMAE bit enables or disables DMA transfer request for common FIFO

The CFDMAE bit cannot be set in GL\_SLEEP or GL\_RESET mode.

Do not enable a DMA transfer for a Common FIFO that is configured as TX FIFO.

This bit is cleared when the CANFD module is in GL\_RESET mode.

**32.2.36 CFDCDTSTS : DMA Transfer Status Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00CC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFDM ASTS	—	—	—	—	—	—	RFD ASTS1	RFD ASTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFDMASTS0	DMA Transfer Status for RX FIFO 0 0: DMA transfer stopped 1: DMA transfer on going	R
1	RFDMASTS1	DMA Transfer Status for RX FIFO 1 0: DMA transfer stopped 1: DMA transfer on going	R
7:2	—	These bits are read as 0.	R
8	CFDMASTS	DMA Transfer Status only for Common FIFO 0: DMA transfer stopped 1: DMA transfer on going	R
31:9	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The DMA Transfer Status Register shows the status of the DMA transfer.

### RFDMASTSe (e = 0 to 1) bit (DMA Transfer Status for RX FIFO e)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty. Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.RFDMAEe (see CFDCDTCT.RFDMAEe bit in [section 32.2.35. CFDCDTCT : DMA Transfer Control Register](#)) is set to 0 while DMA transfer for the corresponding FIFO is on going, the RFDMASTSe bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CANFD module is in GL\_RESET mode.

### CFDMASTS bit (DMA Transfer Status only for Common FIFO)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty. Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.CFDMAE (see CFDCDTCT.CFDMAE bit in [section 32.2.35. CFDCDTCT : DMA Transfer Control Register](#)) is set to 0 while DMA transfer for the corresponding FIFO is on going, the CFDMASTS bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CANFD module is in GL\_RESET mode.

## 32.2.37 CFDTMCI : TX Message Buffer Control Registers i (i = 0 to 3)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0070 + 0x01 × i

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TMOM	TMTA R	TMTR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMTR	TX Message Buffer Transmission Request 0: TX Message buffer transmission not requested 1: TX message buffer transmission requested	R/W
1	TMTAR	TX Message Buffer Transmission Abort Request 0: TX message buffer transmission request abort not requested 1: TX message buffer transmission request abort requested	R/W
2	TMOM	TX Message Buffer One-shot Mode 0: TX message buffer not configured in one-shot mode 1: TX message buffer configured in one-shot mode	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX Message Buffer Control Registers configure the TX message buffer functions.

### TMTR bit (TX Message Buffer Transmission Request)

When the TMTR bit is set, the CANFD module logic tries to transmit the message stored in the corresponding message buffer.

Only write to this bit when the related CANFD module is in CH\_HALT or CH\_OPERATION mode.

Do not set this bit if the corresponding TX message buffer is linked to a COM FIFO in TX mode or is a part of TX Queue.

This bit cannot be directly cleared by a CPU write access.

This bit can only be set when the Transmission Result flag bits (CFDTMSTSj.TMTRF) in the CFDTMSTSj register corresponding to the message buffer are cleared to 00b.

The TMTR bit is automatically cleared by the:

- CANFD module logic at the end of a successful transmission
- CANFD module logic at the end of a transmission abort, requested by the corresponding CFDTMCI.TMTAR bit
- CANFD module logic when there is a detection of a CAN bus error or arbitration loss if CFDTMCI.TMOM bit is set for the message buffer
- CANFD module logic when the CANFD module is in GL\_RESET mode or the related channel is in CH\_RESET mode.

#### TMTAR bit (TX Message Buffer Transmission Abort Request)

When the TMTAR bit is set, the CANFD module logic tries to abort the transmission of the frame stored in the corresponding message buffer.

In most cases, transmission cannot be aborted if the internal scan for transmission is complete and the message buffer has already been selected for transmission. In this case, frame may be transmitted successfully from the message buffer. The message buffer selection is released by entering CH\_HALT mode.

However, message buffer selected for transmission can be aborted by an abort request when the CAN node detects a new message on the bus (RX pin) before it starts transmission from the selected message buffer.

Only write to the TMTAR bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode. This bit can only be set when the related transmit request TMTR bit is set.

The TMTAR bit cannot be cleared by a CPU write access. Clearing of this bit by CANFD has priority over setting by a CPU write access.

The TMTAR bit is automatically cleared by:

- The CANFD module logic at the end of a successful transmission
- The CANFD module logic at the end of a transmission abort
- The CANFD module logic when there is detection of a CAN bus error or arbitration loss
- The CANFD module logic when the CANFD module is in GL\_RESET mode or the related channel enters CH\_RESET mode.

#### TMOM bit (TX Message Buffer One-shot Mode)

When the TMOM bit is set, the CANFD module logic tries to transmit the message only once.

If the transmission is successful, the CFDTMSTsj.TMTRF bits are set to 10b or 11b. Otherwise, the transmission is automatically aborted and CFDTMSTsj.TMTRF bits are set to 01b due to a bus error or a bus arbitration lost.

The TMOM bit remains set if the transmission has completed successfully or aborted due to an error or a loss of arbitration.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

Set this bit at the same time as the TMTR bit. Clear this bit with a write access.

If a message has already been requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.

The TMOM bit is automatically cleared by the CANFD module logic when the CANFD module is in GL\_RESET mode or the related channel is in CH\_RESET mode.

### 32.2.38 CFDTMSTsj : TX Message Buffer Status Registers j (j = 0 to 3)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0074 + 0x01 × j

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TMTA RM	TMTR M	TMTRF[1:0]	TMTS TS	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMTSTS	TX Message Buffer Transmission Status 0: No on-going transmission 1: On-going transmission	R
2:1	TMTRF[1:0]	TX Message Buffer Transmission Result Flag 0 0: No result 0 1: Transmission aborted from the TX message buffer 1 0: Transmission successful from the TX message buffer and transmission abort was not requested 1 1: Transmission successful from the TX message buffer and transmission abort was requested	R/W
3	TMTRM	TX Message Buffer Transmission Request Mirrored 0: TX message buffer transmission not requested 1: TX message buffer transmission requested	R
4	TMTARM	TX Message Buffer Transmission Abort Request Mirrored 0: TX message buffer transmission request abort not requested 1: TX message buffer transmission request abort requested	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX Message Buffer Status Registers show status of the transmission and transmission abort for the corresponding message buffers.

#### **TMTSTS bit (TX Message Buffer Transmission Status)**

The TMTSTS bit is set automatically at the start of the transmission from the corresponding TX message buffer.

This bit is cleared automatically when:

- Transmission stops
- The CANFD module is in GL\_RESET mode
- The related CANFD channel is in CH\_RESET mode.

#### **TMTRF[1:0] bits (TX Message Buffer Transmission Result Flag)**

The TMTRF[1:0] bits show the result for the corresponding TX message buffer. The status is as follows:

- 00: Transmission in progress or has not been requested
- 01: Transmission has been aborted from the corresponding TX message buffer
- 10: Transmission was successful from the corresponding TX message buffer and the CFDTMCi.TMTAR bit was not set for this TX message buffer
- 11: Transmission was successful from the corresponding TX message buffer, but the CFDTMCi.TMTAR bit was set for this TX message buffer.

Only write to these bits when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

The TMTRF[1:0] bits are cleared automatically when the CANFD module is in GL\_RESET mode or the related channel is in CH\_RESET mode.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

#### **TMTRM bit (TX Message Buffer Transmission Request Mirrored)**

The TMTRM bit is set when the CFDTMCi.TMTR bit in the corresponding CFDTMCi register is set.

This bit is cleared when the CFDTMCi.TMTR bit in the corresponding CFDTMCi register is cleared.

#### **TMTARM bit (TX Message Buffer Transmission Abort Request Mirrored)**

The TMTARM bit is set when the CFDTMCi.TMTAR bit in the corresponding CFDTMCi register is set.

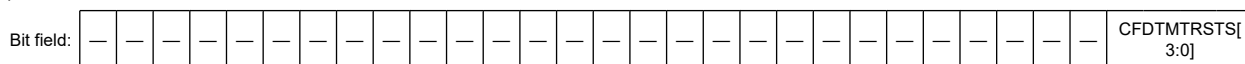
This bit is cleared when the CFDTMCi.TMTAR bit in the corresponding CFDTMCi register is cleared.

### 32.2.39 CFDTMTRSTS : TX Message Buffer Transmission Request Status Register

Base address: CANFD<sub>n</sub> = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFD<sub>n</sub>\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0078

Bit position: 31 3 0



Value after reset: 0

Bit	Symbol	Function	R/W
3:0	CFDTMTRSTS[3:0]	TX Message Buffer Transmission Request Status 0: Transmission not requested for corresponding TX message buffer 1: Transmission requested for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Transmission Request Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTRSTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTRSTS corresponds to the buffer number of TX message buffer.

#### CFDTMTRSTS[3:0] bits (TX Message Buffer Transmission Request Status)

The CFDTMTRSTS[3:0] bits show status of the CFDTMCi.TMTR bits of the TX Message Buffer Control Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers (CFDTMCi), and only when the message buffer does not belong to a TX Queue.

Each bit is cleared automatically when:

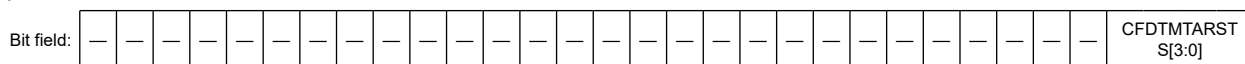
- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CANFD module is in GL\_RESET mode
- The related CANFD channel is in CH\_RESET mode.

### 32.2.40 CFDTMTARSTS : TX Message Buffer Transmission Abort Request Status Register

Base address: CANFD<sub>n</sub> = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFD<sub>n</sub>\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x007C

Bit position: 31 3 0



Value after reset: 0

Bit	Symbol	Function	R/W
3:0	CFDTMTARSTS[3:0]	TX Message Buffer Transmission Abort Request Status 0: Transmission abort not requested for corresponding TX message buffer 1: Transmission abort requested for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Transmission Abort Request Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTARSTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTARSTS corresponds to the buffer number of TX message buffer.

**CFDTMTARSTS[3:0] bits (TX Message Buffer Transmission Abort Request Status)**

The CFDTMTARSTS[3:0] bits show status of the CFDTMCI.TMTAR bits of the TX Message Buffer Control Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers, and when the message buffer belongs to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CANFD module is in GL\_RESET mode
- The related CANFD channel is in CH\_RESET mode.

**32.2.41 CFDTMTCSTS : TX Message Buffer Transmission Completion Status Register**

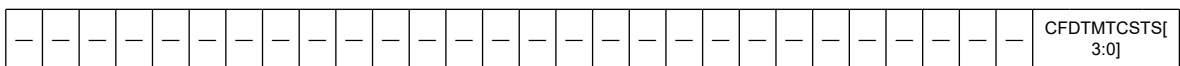
Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0080

Bit position: 31

3 0

Bit field:



Value after reset: 0

Bit	Symbol	Function	R/W
3:0	CFDTMTCSTS[3:0]	TX Message Buffer Transmission Completion Status 0: Transmission not complete for corresponding TX message buffer 1: Transmission completed for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Transmission Completion Status for the corresponding TX Message Buffer. The bit 0 of a CFDTMTCSTS register corresponds to the TX message buffer 0.

The bit position of CFDTMTCSTS corresponds to the buffer number of TX message buffer.

**CFDTMTCSTS[3:0] bits (TX Message Buffer Transmission Completion Status)**

The CFDTMTCSTS[3:0] bits show status of successful completion of the TX Message Buffer Status Registers.

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Status Registers.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Status Registers
- The CANFD module is in GL\_RESET mode
- The related CANFD channel is in CH\_RESET mode.

If a CAN channel enters CH\_RESET mode, then the bits related to that channel are cleared.

**32.2.42 CFDTMTASTS : TX Message Buffer Transmission Abort Status Register**

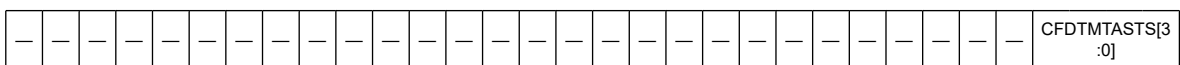
Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0084

Bit position: 31

3 0

Bit field:



Value after reset: 0

Bit	Symbol	Function	R/W
3:0	CFDGMTASTS[3:0]	TX Message Buffer Transmission Abort Status 0: Transmission not aborted for corresponding TX message buffer 1: Transmission aborted for corresponding TX message buffer	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Transmission abort Status for the corresponding TX Message Buffer. The bit 0 of a CFDGMTASTS register corresponds to the TX message buffer 0.

The bit position of CFDGMTASTS corresponds to the buffer number of TX message buffer.

**CFDGMTASTS[3:0] bits (TX Message Buffer Transmission Abort Status)**

The CFDGMTASTS[3:0] bits show status of the successful transmission abort of the corresponding TX message buffer.

Each bit is set automatically when the CFDMSTsj.TMTRF bits are set to 01b in the corresponding TX Message Buffer Status Register.

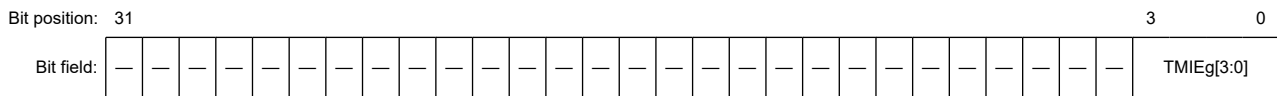
Each bit is cleared automatically when:

- The CFDMSTsj.TMTRF bits are cleared in the corresponding TX Message Buffer Status Register
- The CANFD module is in GL\_RESET mode
- The related CANFD channel is in CH\_RESET mode.

**32.2.43 CFDMIEC : TX Message Buffer Interrupt Enable Configuration Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0088



Value after reset: 0

Bit	Symbol	Function	R/W
3:0	TMIEg[3:0]	TX Message Buffer Interrupt Enable 0: TX message buffer interrupt disabled for corresponding TX message buffer 1: TX message buffer interrupt enabled for corresponding TX message buffer	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

These bits show the TX Message Buffer Interrupt Enable for the corresponding TX Message Buffer.

The bit 0 of a CFDMIEC register corresponds to the TX message buffer 0.

The bit position of CFDMIEC corresponds to the buffer number of TX message buffer.

g = [0...3]

**TMIEg[3:0] bits (TX Message Buffer Interrupt Enable)**

If the TMIEg[3:0] bits are set, an interrupt is generated at the end of a successful transmission from the corresponding message buffer.

See [section 32.7. Interrupts and DMA](#) for TX Message Buffer Interrupt specification.

Do not write to the TMIEg[7:0] bits when:

- The CANFD module is in GL\_SLEEP mode
- The related CANFD channel is in CH\_SLEEP mode
- The corresponding TX message buffer is linked to a Common FIFO with the CFDCFCC.CFTML bits.



### 32.2.44 CFDTXQCC : TX Queue Configuration/Control Register

Base address: CANFD<sub>n</sub> = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFD<sub>n</sub>\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x008C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TXQDC[1:0]	TXQIM	—	TXQTXIE	—	—	—	—	—	TXQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled	R/W
4:1	—	These bits are read as 0. The write value should be 0.	R/W
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX interrupt disabled 1: TX Queue TX interrupt enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TXQIM	TX Queue Interrupt Mode 0: When the last message is successfully transmitted 1: At every successful transmission	R/W
9:8	TXQDC[1:0]	TX Queue Depth Configuration 0x00: 0 messages 0x01: Reserved 0x10: 3 messages 0x11: 4 messages	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX Queue Configuration/Control Registers are used to configure the TX Queue transmission.

TXQ is composed of TXMB0 to TXMB3 (at the maximum) when TXQE is enabled.

#### TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC.TXQDC == 0x00).

You cannot write to this bit when the CANFD module is in GL\_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH\_RESET or CH\_SLEEP mode.

The TXQE bit is cleared automatically when the related CANFD channel is in CH\_RESET mode.

#### TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CANFD module is in GL\_SLEEP mode.

Do not write to this bit when the related CANFD channel is in CH\_SLEEP mode.

#### TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CANFD module is in GL\_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH\_SLEEP
- CH\_HALT
- CH\_OPERATION.

**TXQDC[1:0] bits (TX Queue Depth Configuration)**

The TXQDC[1:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[0] up to MB[3] depending on the configured depth.

You cannot write to this bit when the CANFD module is in GL\_SLEEP mode.

Do not write to this bit when the related CANFD channel is in any of the following modes:

- CH\_SLEEP
- CH\_HALT
- CH\_OPERATION.

**32.2.45 CFDTXQSTS : TX Queue Status Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TXQMC[2:0]			—	—	—	—	—	TXQT XIF	TXQF LL	TXQE MP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TXQEMP	TX Queue Empty 0: TX Queue not empty 1: TX Queue empty	R
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full	R
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt condition not satisfied after a frame TX 1: TX Queue interrupt condition satisfied after a frame TX	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	TXQMC[2:0]	TX Queue Message Count Number of messages in the TX Queue.	R
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX Queue Status Registers show the status of the TX Queue of corresponding CAN channel.

**TXQEMP bit (TX Queue Empty)**

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CANFD channel is in CH\_RESET mode.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

**TXQFLL bit (TX Queue Full)**

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CANFD channel is in CH\_RESET mode.

**TXQTXIF bit (TX Queue TX Interrupt Flag)**

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

You cannot write to this bit when the related CANFD channel is in CH\_SLEEP or CH\_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH\_RESET mode.

**TXQMC[2:0][13:8] bits (TX Queue Message Count)**

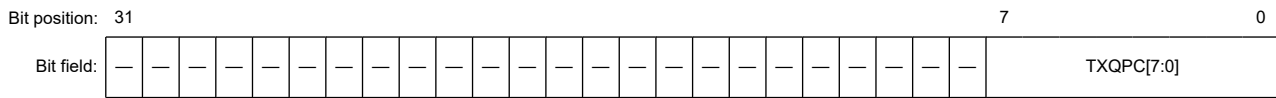
The TXQMC[2:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CANFD channel is in CH\_RESET mode.

**32.2.46 CFTXQPCTR : TX Queue Pointer Control Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0094



Value after reset: 0

Bit	Symbol	Function	R/W
7:0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The TX Queue Pointer Control Registers are used to confirm storage of a full message in the corresponding TX Queue buffers.

**TXQPC[7:0] bits (TX Queue Pointer Control)**

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. Do not write to the FIFO control registers when DMA is enabled.

You cannot write to these bits when the related CANFD channel is in CH\_SLEEP or CH\_RESET mode.

Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full

- The Common FIFO is enabled.

### 32.2.47 CFDTLCC : TX History List Configuration/Control Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0098

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	THLD TE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	THLE	TX History List Enable 0: TX History List disabled 1: TX History List enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	THLIE	TX History List Interrupt Enable 0: TX History List Interrupt disabled 1: TX History List Interrupt enabled	R/W
9	THLIM	TX History List Interrupt Mode 0: Interrupt generated if TX History List level reaches ¾ of the TX History List depth 1: Interrupt generated for every successfully stored entry	R/W
10	THLDTE	TX History List Dedicated TX Enable 0: TX FIFO + TX Queue 1: Flat TX MB + TX FIFO + TX Queue	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX History List Configuration/Control Register configures the TX History List functions.

#### THLE bit (TX History List Enable)

The THLE bit enables the TX History List buffer when it is set.

You cannot write to this bit when the related CANFD channel is in CH\_RESET or CH\_SLEEP mode.

This bit is cleared automatically when the related CANFD channel is in CH\_RESET mode.

#### THLIE bit (TX History List Interrupt Enable)

The THLIE bit enables the generation of the TX History List interrupt when it is set.

You cannot write to this bit when the CANFD module is in GL\_SLEEP mode.

#### THLIM bit (TX History List Interrupt Mode)

The THLIM bit selects the interrupt generation condition for the FIFO.

You cannot write to this bit when the CANFD module is in GL\_SLEEP mode.

Do not write to this bit when the CANFD module is in GL\_HALT or GL\_OPERATION mode.

#### THLDTE bit (TX History List Dedicated TX Enable)

The THLDTE bit selects the condition for storing an entry in the TX History List after successful transmission.

You cannot write to this bit when the CANFD module is in GL\_SLEEP mode.

Do not write to this bit when the CANFD module is in GL\_HALT or GL\_OPERATION mode.

### 32.2.48 CFDTHLSTS : TX History List Status Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x009C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	THLMC[3:0]				—	—	—	—	THLIF	THLELT	THLFL	THLEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	THLEMP	TX History List Empty 0: TX History List not empty 1: TX History List empty	R
1	THLFL	TX History List Full 0: TX History List not full 1: TX History List full	R
2	THLELT	TX History List Entry Lost 0: No entry lost in TX History List 1: TX History List entry Lost	R/W
3	THLIF	TX History List Interrupt Flag 0: TX History List interrupt condition not satisfied 1: TX History List interrupt condition satisfied	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
11:8	THLMC[3:0]	TX History List Message Count Number of messages stored in TX History List	R
31:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The TX History List Status register shows the status of data stored in the TX History List buffer.

#### THLEMP bit (TX History List Empty)

The THLEMP bit is set automatically when the CPU has read all the entries from the TX History List buffer.

This bit is cleared automatically when the first entry is stored to the TX History List.

This bit is set automatically when:

- TX History List is disabled
- The related CANFD channel is in CH\_RESET mode.

#### THLFL bit (TX History List Full)

The THLFL bit is set automatically when the number of entries in the TX History List buffer matches the TX History List depth.

Each TX History List can store up to 8 entries.

This bit is cleared automatically when:

- The number of entries in the TX History List buffer is less than the TX History List depth
- The TX History List is disabled
- The related CANFD channel is in CH\_RESET mode.

**THLELT bit (TX History List Entry Lost)**

The THLELT bit is set when a new entry cannot be stored because the related TX History List buffer is already full.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH\_RESET mode.

**THLIF bit (TX History List Interrupt Flag)**

The THLIF bit is set when the configured interrupt condition is satisfied.

Only write to this bit when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If the set from the CAN channel occurs simultaneously with the clear by the write access, then the bit is set.

This bit is cleared:

- By writing 0 to it
- When the related CANFD channel is in CH\_RESET mode.

The bit is cleared by writing 0 to it.

This bit is automatically cleared in CH\_RESET mode.

**THLMC[3:0] bits (TX History List Message Count)**

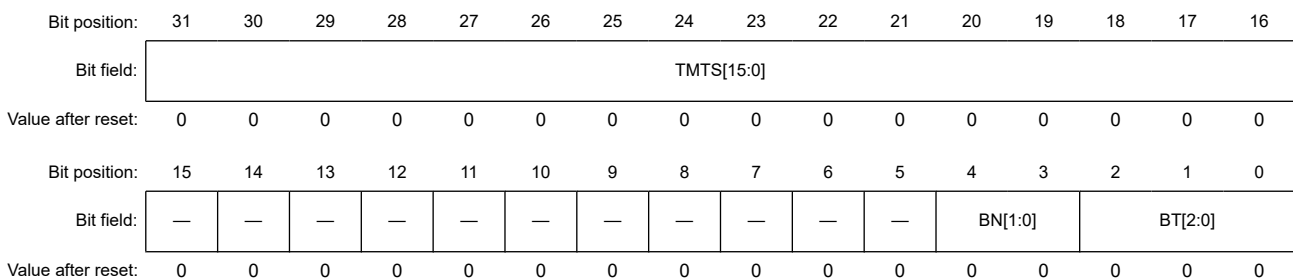
The THLMC[3:0] bits show the number of transmitted messages stored in the TX History List.

These bits are cleared automatically when the related CANFD channel is in CH\_RESET mode.

**32.2.49 CFDTHLACC0 : TX History List Access Register 0**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0740



Bit	Symbol	Function	R/W
2:0	BT[2:0]	Buffer Type 0 0 1: Flat TX message buffer 0 1 0: TX FIFO message buffer number 1 0 0: TX Queue message buffer number	R
4:3	BN[1:0]	Buffer Number Number of the message buffer	R

Bit	Symbol	Function	R/W
15:5	—	These bits are read as 0.	R
31:16	TMTS[15:0]	Transmit Timestamp Transmit timestamp value for software drivers	R

Note: S-TYPE-3, P-TYPE-3

The TX History List Access Registers 0 provide access to the entry in the TX History List based on the read timestamp value.

**BT[2:0] bits (Buffer Type)**

The BT[2:0] bits indicate whether data has been stored following a transmission from a FIFO buffer, a TX Queue or a TX message buffer.

**BN[1:0] bits (Buffer Number)**

The BN[1:0] bits show the message buffer from which transmission was successfully completed. If a message from a Common FIFO is transmitted, then these bits show the message buffer that is linked to the Common FIFO for transmission.

**TMTS[15:0] bits (Transmit Timestamp)**

The TMTS[15:0] bits indicate the timestamp for use by software drivers.

**32.2.50 CFDTLACC1 : TX History List Access Register 1**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0744

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TIFL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TID[15:0]	Transmit ID These bits indicate that message buffer reference ID, TX FIFO reference ID, or AFL pointer field is stored for software drivers.	R
17:16	TIFL[1:0]	Transmit Information Label These bits indicate that message buffer information label, TX FIFO information label, or AFL information label is stored for software drivers.	R
31:18	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

The TX History List Access Registers 1 provide access to entry in the TX History List based on the read pointer value.

**TID[15:0] bits (Transmit ID)**

The TID[15:0] bits indicate whether the message buffer reference ID (CFDTMFDCTRb.TMPTR) or the TX FIFO reference ID (CFDCFFDCSTS.CFPTR) is for use by software drivers.

**TIFL[1:0] bits (Transmit Information Label)**

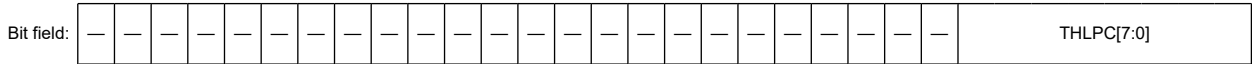
The TIFL[1:0] bits indicate whether the message buffer information label (CFDTMFDCTRb.TMIFL) or the TX FIFO information label (CFDCFFDCSTS.CFIFL) is for use by software drivers.

### 32.2.51 CFDTLPCCTR : TX History List Pointer Control Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00A0

Bit position: 31 7 0



Value after reset: 0

Bit	Symbol	Function	R/W
7:0	THLPC[7:0]	TX History List Pointer Control Increments the write pointer to the TX History List in the corresponding channel	W
31:8	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The TX History List Pointer Control Registers are used to increment the read pointer of the TX History List.

#### THLPC[7:0] bits (TX History List Pointer Control)

When 0xFF is written to the THLPC[7:0] bits, the read pointer of the TX History List is moved to the next TX History List entry address.

The read value from these bits is always 0x00. Only write to these bits when the related CANFD channel is in CH\_HALT or CH\_OPERATION mode.

Only write 0xFF to these registers when the corresponding TX History List is enabled and not empty.

### 32.2.52 CFDGRSTC : Global SW reset Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00D8

Bit position: 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	SRST	SW Reset 0: Normal state 1: SW reset state	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits control the validity of rewriting of a SRST bit.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

#### SRST bit (SW Reset)

When the SRST bit is set, the CANFD module is in the same state as hardware reset. When a reset is required, write 1 then write 0 to this bit.

This bit is cleared when the CANFD module is in GL\_SLEEP mode.



When this bit is cleared, the RAM initialization sequence does not operate. The configuration of RAM is performed by software.

The RAM is not initialized when software reset is performed during the initialization of RAM. Software must perform the initialization of RAM.

**KEY[7:0] bits (Key Code)**

When 0xC4 is written in the KEY[15:8] bits, a write to the SRST bit is valid.

The read value from these bits is always 0x00.

CFDGRSTC.SRST bit and the CFDGRSTC.KEY bit should be written simultaneously.

**32.2.53 CFDTSTCFG : Global Test Configuration Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00A8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	RTMPS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
19:16	RTMPS[3:0]	RAM Test Mode Page Select Select a RAM test mode page	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Test Configuration Register is used to configure the RAM test mode page.

**RTMPS[3:0] bits (RAM Test Mode Page Select)**

The RTMPS[3:0] bits select the RAM page mode for CPU read/write access when the CANFD module is configured in RAM test mode.

See [section 32.9.2.1. RAM Test Mode](#) for the RAM test mode specification.

Do not write to these bits when the CANFD module is in GL\_RESET or GL\_SLEEP mode.

Only enter values from 0 to 8 (0x008) for the message buffer RAM.

Only write to these bits when the CANFD module is in GL\_HALT mode.

These bits are cleared automatically when the related CANFD channel is in GL\_RESET mode.

### 32.2.54 CFDGTSTCTR : Global Test Control Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00AC

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

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
2	RTME	RAM Test Mode Enable 0: RAM test mode disabled 1: RAM test mode enabled	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The Global Test Control register is used to control the global test modes of the CANFD module.

#### RTME bit (RAM Test Mode Enable)

When the RTME bit is set, the CANFD module is configured in RAM test mode. See [section 32.9.2.1. RAM Test Mode](#) for RAM test mode specification.

Only write to this bit when the CANFD module is in GL\_HALT mode.

Clear this bit when the CANFD module is in GL\_HALT mode.

This bit is cleared automatically when the CANFD module is in GL\_RESET mode.

### 32.2.55 CFDFDCFG : Global FD Configuration Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00B0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TSCCFG[1:0]	—	—	—	—	—	—	—	—	RPED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RPED	RES Bit Protocol Exception Disable 0: Protocol exception event detection enabled 1: Protocol exception event detection disabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
9:8	TSCCFG[1:0]	Timestamp Capture Configuration 0 0: Timestamp capture at the sample point of SOF (start of frame) 0 1: Timestamp capture at frame valid indication 1 0: Timestamp capture at the sample point of RES bit 1 1: Reserved	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

### RPED bit (RES Bit Protocol Exception Disable)

The RPED bit configures the protocol exception event handling according to ISO 11898-1.

When this bit is enabled, the protocol exception event detection is disabled, and the protocol controller transmits an error frame when the protocol exception event is detected (RES bit is sampled recessive).

Only write to this bit when the CANFD module is in GL\_RESET mode.

### TSCCFG[1:0] bits (Timestamp Capture Configuration)

The TSCCFG[1:0] bits configure the different capture points of the timestamp for transmission and reception.

When  $CFDGFDCFG.TSCCFG[1:0] = 10b$ , the timestamp capture is performed for CANFD frames at RES bit and for Classical frames at the start of frame.

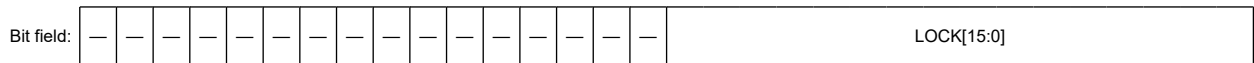
Only write to these bits when the CANFD module is in GL\_RESET mode.

## 32.2.56 CFDGLOCKK : Global Lock Key Register

Base address:  $CANFD_n = 0x4038\_0000 + 0x2000 \times n$  ( $n = 0, 1$ )  
 $CANFD_n\_NS = 0x5038\_0000 + 0x2000 \times n$  ( $n = 0, 1$ )

Offset address: 0x00B8

Bit position: 31 15 0



Value after reset: 0

Bit	Symbol	Function	R/W
15:0	LOCK[15:0]	Lock Key Key bits for unlocking the protection of test modes	W
31:16	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The Global Lock Key register is a write-only register that is used to unlock the protection for special test bits.

See [section 32.9.2. Global Test Modes](#) for Lock key specification.

### LOCK[15:0] bits (Lock Key)

The unlock key sequence must be written in the LOCK[15:0] bits to configure the CANFD module in RAM test modes.

The read value from these bits is always 0x0000.

You cannot write to these bits when the CANFD module is in GL\_SLEEP or GL\_RESET mode.

Do not write to these bits when the CANFD module is in GL\_OPERATION mode.



### 32.2.59 CFDGAFIGNCTR : Global AFL Ignore Control Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x00C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	IREN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IREN	Ignore Rule Enable 0: AFL entry number is not ignored 1: AFL entry number is ignored	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits control the validity of rewriting the IREN bit.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

#### IREN bit (Ignore Rule Enable)

When the IREN bit is set, the entry number (selected by CFDGAFIGNENT register) is ignored.

This bit is cleared automatically when the CANFD module is in GL\_RESET mode.

#### KEY[7:0] bits (Key Code)

When 0xC4 is written in the KEY[7:0] bits, a write to the IREN bit is valid.

The read value from these bits is always 0x00.

CFDGAFIGNCTR.IREN bit and the CFDGAFIGNCTR.KEY bit should be written simultaneously

### 32.2.60 CFDRMIEC : RX Message Buffer Interrupt Enable Configuration Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0038

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RMIE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RMIE[15:0]	RX Message Buffer Interrupt Enable 0: RX Message Buffer Interrupt disabled for corresponding RX message buffer 1: RX Message Buffer Interrupt enabled for corresponding RX message buffer	R/W

Note: S-TYPE-3, P-TYPE-3

These bits show the RX Message Buffer Interrupt Enable for the corresponding RX Message Buffer. CFDRMIEC bit 0 corresponds to RX Message Buffer 0 and so on.

The bit position of CFDRMIEC corresponds to the buffer number of RXMB.

**RMIE[15:0] bit (RX Message Buffer Interrupt Enable)**

If this bit is set, then an interrupt will be generated at the end of a successful reception from the corresponding Message Buffer.

For details, see [section 32.7.1. Interrupts](#).

Users cannot write to this bit when the CANFD module is in GL\_SLEEP mode.

**32.2.61 Message Buffer Component Structure****32.2.61.1 Start Addresses**

The start address for each of the Message Buffer component is calculated using the number of related Message Buffer components.

The start addresses for each register in the Message Buffer component are depicted in [Table 32.4](#).

**Table 32.4 Message Buffer Component Register Start Addresses**

<b>b = Message buffer component index</b>	<b>MBCP</b>	<b>p</b>	<b>Register</b>	<b>Start Address</b>
[0...15] b = [0...7]	RMBCPb[0]	x	RMID	0x0920 + b × 0x004C
		x	RMPTR	0x0924 + b × 0x004C
		x	RMFDSTS b	0x0928 + b × 0x004C
		[1...15]	RMDFBp	0x092C + b × 0x004C + p × 0x0004
[0...15] b = [8...15]	RMBCPb[0]	x	RMIDb	0x0D20 + (b-8) × 0x004C
		x	RMPTRb	0x0D24 + (b-8) × 0x004C
		x	RMFDSTS b	0x0D28 + (b-8) × 0x004C
		[1...15]	RMDFBp	0x0D2C + (b-8) × 0x004C + p × 0x0004
[0...1]	RFMBCPb[0]	x	RFIDb	0x0520 + b × 0x004C
		x	RFPTRb	0x0524 + b × 0x004C
		x	RFFDSTS b	0x0528 + b × 0x004C
		[1...15]	RFDFBp	0x052C + b × 0x004C + p × 0x0004
[0]	CFMBCPb[0]	x	CFID	0x05B8
		x	CFPTR0	0x05BC
		x	CFFDCST S0	0x05C0
		[1...15]	CFDFp0	0x05C4 + p × 0x0004
[0...3]	TMBCPb[0]	x	TMIDb	0x0604 + b × 0x004C
		x	TMPTRb	0x0608 + b × 0x004C
		x	TMFDCTR b	0x060C + b × 0x004C
		[1...15]	TMDFBp	0x0610 + b × 0x004C + p × 0x0004

The message buffer configuration consists of four types of Message Buffer components:

- RX Message Buffer Component (CFDRMBCPb[0])
- RX FIFO Access Message Buffer Component (CFDRFMBCPb[0])
- Common FIFO Access Message Buffer Component (CFDCFMBCP0[0])
- TX Message Buffer Component (CFDTMBCPb[0]).

Where b = the Message Buffer component index that has a range that varies based on the type of Message Buffer component.

For a summary of this configuration, see [Figure 32.28](#). For a detailed description of the number of and the different types of message buffers, see [section 32.6. FIFO Buffers and Normal Message Buffer Configuration](#).

As described in [section 32.2. Register Descriptions](#), each Message Buffer component consists of the following registers:

- Identifier (ID)
- Pointer (PTR)
- Data Field (DFp).

Where p = the Data Field register index that has a range that varies based on the type of message buffer component.

Rc is the Message Buffer Component register where c = Message Buffer Component register index that has a range that varies based on the type of Message Buffer component.

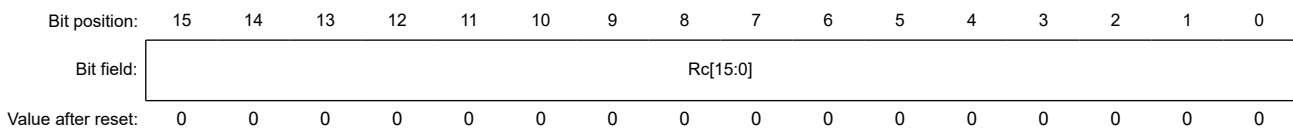
A description of the registers, their associated bits and their accessibility are shown below the summary and detailed figures of each component.

In each of the figures, a cell that contains ‘-‘ means reserved and has the same behavior as reserved bits for registers in [section 32.2.61. Message Buffer Component Structure](#).

### 32.2.61.2 CFDRMBCPb[0] : RX Message Buffer Component b (b = 0 to 15)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: See [Table 32.4](#)



Bit	Symbol	Function	R/W
15:0	Rc[15:0]	RX Message Buffer Component c Refer to <a href="#">Table 32.5</a> , <a href="#">Table 32.6</a> and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R/W

Note: S-TYPE-3, P-TYPE-3

Where the total number of CFDRMBCPb = 16 as shown in [Figure 32.28](#) (c = RX Message Buffer Component Register index = [0...18])

#### Rc[15:0] bit (RX Message Buffer Component c)

The RX Message Buffer Component is made up of the following registers: CFDRMIDb, CFDRMPTRb, CFDRMFDSTsb, and CFDRMDFbp. Refer to [Table 32.6](#) for details of how to interpret the structure of this buffer component and how to access the respective registers.

**Table 32.5 RX Message Buffer Component Summary (1 of 2)**

RX Message Buffer Component (RMBCP)	
Rc	CANFD mode (CAN_FD_MODE = '1'b1)
R0	RX Message Buffer (b) ID Registers
R1	RX Message Buffer (b) Pointer Registers
R2	RX Message Buffer (b) CANFD Status Registers
R3	RX Message Buffer (b) Data Field 0 Registers
R4	RX Message Buffer (b) Data Field 1 Registers
R5	RX Message Buffer (b) Data Field 2 Registers
R6	RX Message Buffer (b) Data Field 3 Registers

**Table 32.5 RX Message Buffer Component Summary (2 of 2)**

RX Message Buffer Component (RMBCP)	
Rc	CANFD mode (CAN_FD_MODE = 1'b1)
R7	RX Message Buffer (b) Data Field 4 Registers
R8	RX Message Buffer (b) Data Field 5 Registers
R9	RX Message Buffer (b) Data Field 6 Registers
R10	RX Message Buffer (b) Data Field 7 Registers
R11	RX Message Buffer (b) Data Field 8 Registers
R12	RX Message Buffer (b) Data Field 9 Registers
R13	RX Message Buffer (b) Data Field 10 Registers
R14	RX Message Buffer (b) Data Field 11 Registers
R15	RX Message Buffer (b) Data Field 12 Registers
R16	RX Message Buffer (b) Data Field 13 Registers
R17	RX Message Buffer (b) Data Field 14 Registers
R18	RX Message Buffer (b) Data Field 15 Registers
R[19...31]	—

**Table 32.6 RX Message Buffer Component (RMBCP) Detailed**

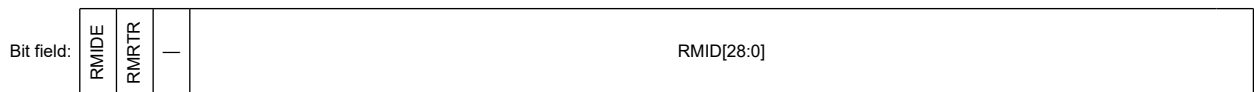
Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
R0	x	CFDRMI Db	RMIDE	RMRTR	—	RMID																																		
R1	x	CFDRM PTRb	RMDLC			—	—	—	—	—	—	—	—	—	—	—	—	RMITS																						
R2	x	CFDRM FDSTSb	RMPTR															—	—	—	—	—	—	RMIFL	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R3	0	CFDRM DFbp	RMDB_HH					RMDB_HL					RMDB_LH					RMDB_LL																						
R[4...18]	[1...15]	CFDRM DFbp	RMDB_HH					RMDB_HL					RMDB_LH					RMDB_LL																						

**32.2.61.3 CFDRMIDb : RX Message Buffer ID Registers (b = 0 to 15)**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0920 + 0x004C × b (b = 0 to 7)  
 0x0D20 + 0x004C × (b - 8) (b = 8 to 15)

Bit position: 31 30 29 0



Value after reset: 0

Bit	Symbol	Function	R/W
28:0	RMID[28:0]	RX Message Buffer ID Field STD-ID/EXT-ID fields	R
29	—	This bit is read as 0.	R



Bit	Symbol	Function	R/W
30	RMRTR	RX Message Buffer RTR Bit 0: Data frame 1: Remote frame	R
31	RMIDE	RX Message Buffer IDE Bit 0: STD-ID is stored 1: EXT-ID is stored	R

Note: S-TYPE-3, P-TYPE-3

The RX Message Buffer ID Register b (b = 0 to 15) store the ID field, IDE bit, and RTR bit of the received message.

**RMID[28:0] bits (RX Message Buffer ID Field)**

The RMID[28:0] are the bits of the STD-ID/EXT-ID fields of the message stored in the RX message buffer.

See section 32.2.61.1. Start Addresses for details on how to interpret the structure of this buffer component.

**RMRTR bit (RX Message Buffer RTR Bit)**

The RMRTR bit shows whether a data frame or a remote frame was stored in the RX message buffer.

Note: There are no remote frames in CANFD format. When a CANFD frame is received, the register reflects the state of the received value (the RRS bit in FD frame format).

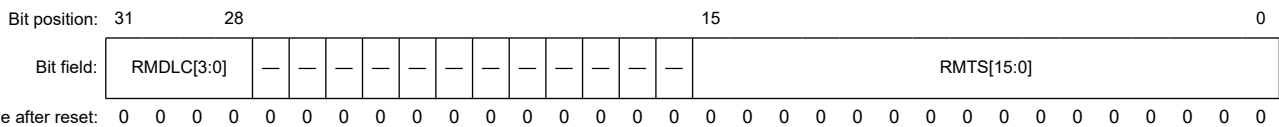
**RMIDE bit (RX Message Buffer IDE Bit)**

The RMIDE bit shows whether message with Standard Identifier or Extended Identifier was stored in the RX message buffer.

**32.2.61.4 CFDRMPTRb : RX Message Buffer Pointer Registers (b = 0 to 15)**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0924 + 0x004C × b (b = 0 to 7)  
0x0D24 + 0x004C × (b - 8) (b = 8 to 15)



Bit	Symbol	Function	R/W
15:0	RMTS[15:0]	RX Message Buffer Timestamp Field Timestamp value stored for the message in the RX message buffer	R
27:16	—	These bits are read as 0.	R
31:28	RMDLC[3:0]	RX Message Buffer DLC Field Number of data bytes received in a CAN frame.	R

Note: S-TYPE-3, P-TYPE-3

The RX Message Buffer Pointer Register b (b = 0 to 15) store the DLC and Timestamp fields for the received message.

**RMTS[15:0] bits (RX Message Buffer Timestamp Field)**

The RMTS[15:0] bits store the timestamp value taken at the capture point as configured by CFDFGDCFG.TSCCFG of the received message.

**RMDLC[3:0] bits (RX Message Buffer DLC Field)**

The RMDLC[3:0] bits store the number of data bytes that were received in the RX message buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

Note: The maximum capacity of the buffer belongs to CFDRMNB.RMPLS and this is not available in the classical CAN function.

### 32.2.61.5 CFDRMFDSTSb : RX Message Buffer CANFD Status Registers (b = 0 to 15)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0928 + 0x004C × b (b = 0 to 7)  
 0x0D28 + 0x004C × (b - 8) (b = 8 to 15)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RMPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RMIFL[1:0]	—	—	—	—	—	RMFD F	RMBR S	RMESI	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RMESI <sup>*1</sup>	Error State Indicator bit 0: CANFD frame received from error active node 1: CANFD frame received from error passive node	R
1	RMBRS <sup>*1</sup>	Bit Rate Switch bit 0: CANFD frame received with no bit rate switch 1: CANFD frame received with bit rate switch	R
2	RMFDF <sup>*1</sup>	CAN FD Format bit 0: Non CANFD frame received 1: CANFD frame received	R
7:3	—	These bits are read as 0.	R
9:8	RMIFL[1:0]	RX Message Buffer Information Label Field	R
15:10	—	These bits are read as 0.	R
31:16	RMPTR[15:0]	RX Message Buffer Pointer Field	R

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is not available in the classical CAN function.

The RX Message Buffer CANFD Status Register b (b = 0 to 15) show the status of the FDF, BRS and ESI bits, and pointer of the received CANFD frame.

#### RMESI bit (Error State Indicator bit)

The RMESI bit has the same value as the ESI bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

#### RMBRS bit (Bit Rate Switch bit)

The RMBRS bit has the same value as the BRS bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

#### RMFDF bit (CAN FD Format bit)

The RMFDF bit has the same value as the FDF bit of the received CANFD frame.

Note: This bit is not available in the classical CAN function.

#### RMIFL[1:0] bits (RX Message Buffer Information Label Field)

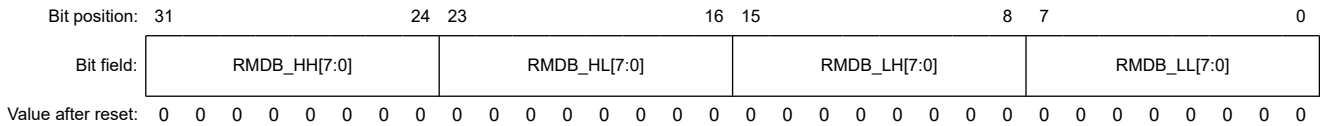
The RMIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

**RMPTR[15:0] bits (RX Message Buffer Pointer Field)**

The RMPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

**32.2.61.6 CFDRMDFb\_p : RX Message Buffer Data Field p Registers (p = 0 to 15, b = 0 to 15)**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)  
 Offset address: 0x092C + 0x004C × b + 0x0004 × p (b = 0 to 7, p = 0 to 15)  
 0x0D2C + 0x004C × (b - 8) + 0x0004 × p (b = 8 to 15, p = 0 to 15)



Bit	Symbol	Function	R/W
7:0	RMDB_LL[7:0]	RX Message Buffer Data Byte (p × 4)	R
15:8	RMDB_LH[7:0]	RX Message Buffer Data Byte ((p × 4) + 1)	R
23:16	RMDB_HL[7:0]	RX Message Buffer Data Byte ((p × 4) + 2)	R
31:24	RMDB_HH[7:0]*1	RX Message Buffer Data Byte ((p × 4) + 3)	R

Note: S-TYPE-3, P-TYPE-3  
 Note 1. These bits are not available in the classical CAN function.

The RX Message Buffer Data Field p Register b (p = 0 to 15, b = 0 to 15) store the data bytes (p × 4) to data bytes ((p × 4) + 3) of the received message.

**RMDB\_LL[7:0] bits (RX Message Buffer Data Byte (p × 4))**

The RMDB\_LL[7:0] bits store data bytes (p × 4) of the message in the RX message buffer.  
 Unused data bytes are filled with 0x00.

**RMDB\_LH[7:0] bits (RX Message Buffer Data Byte ((p × 4) + 1))**

The RMDB\_LH[7:0] bits store data bytes ((p × 4) + 1) of the message in the RX message buffer.  
 Unused Data Bytes will be filled with 0x00.

**RMDB\_HL[7:0] bits (RX Message Buffer Data Byte ((p × 4) + 2))**

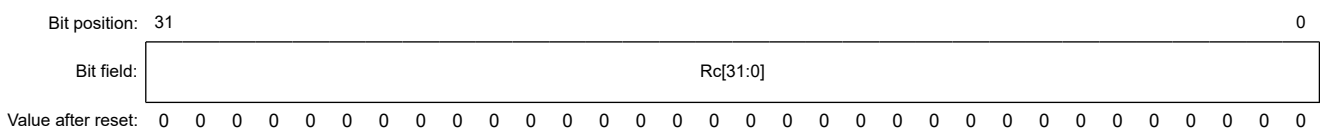
The RMDB\_HL[7:0] bits store data bytes ((p × 4) + 2) of the message in the RX message buffer.  
 Unused data bytes are filled with 0x00.

**RMDB\_HH[7:0] bits (RX Message Buffer Data Byte ((p × 4) + 3))**

The RMDB\_HH[7:0] bits store data bytes ((p × 4) + 3) of the message in the RX message buffer.  
 Unused data bytes are filled with 0x00.

**32.2.61.7 CFDRFMBCPb[0] : RX FIFO Access Message Buffer Component b (b = 0 to 1)**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)  
 Offset address: see [Table 32.4](#)



Bit	Symbol	Function	R/W
31:0	Rc[31:0]	RX FIFO Access Message Buffer Component c See <a href="#">Table 32.7</a> , <a href="#">Table 32.8</a> and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R

Note: S-TYPE-3, P-TYPE-3

Where the total number of CFDRFMBCPb = 2 as shown in [Figure 32.28](#) (c = RX FIFO Access Message Buffer Component Register index = [0...18])

**Rc[31:0] bits (RX FIFO Access Message Buffer Component c)**

The RX FIFO Access Message Buffer component comprises of the following registers:

- CFDRFIDb
- CFDRFPTRb
- CFDRFFDSTSb
- CFDRFDFbp

See [Table 32.8](#) for details on how to interpret the structure of this buffer component and how to access the respective registers.

**Table 32.7 RX FIFO Access Message Buffer component summary**

Rc	
R0	RX FIFO Access ID Registers
R1	RX FIFO Access Pointer Register
R2	RX FIFO Access CANFD Status Registers
R3	RX FIFO Access Data Field 0 Registers
R4	RX FIFO Access Data Field 1 Registers
R5	RX FIFO Access Data Field 2 Registers
R6	RX FIFO Access Data Field 3 Registers
R7	RX FIFO Access Data Field 4 Registers
R8	RX FIFO Access Data Field 5 Registers
R9	RX FIFO Access Data Field 6 Registers
R10	RX FIFO Access Data Field 7 Registers
R11	RX FIFO Access Data Field 8 Registers
R12	RX FIFO Access Data Field 9 Registers
R13	RX FIFO Access Data Field 10 Registers
R14	RX FIFO Access Data Field 11 Registers
R15	RX FIFO Access Data Field 12 Registers
R16	RX FIFO Access Data Field 13 Registers
R17	RX FIFO Access Data Field 14 Registers
R18	RX FIFO Access Data Field 15 Registers
R[19...31]	—

**Table 32.8 RX Message Buffer Component (RMBCP) Detailed (1 of 2)**

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0			
R0	x	CFDRMI Db	RMIDE	RMFTR	—	RMID																															

**Table 32.8 RX Message Buffer Component (RMBCP) Detailed (2 of 2)**

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
R1	x	CFDRM PTRb	RMDLC					—	—	—	—	—	—	—	—	—	—	—	RMST																
R2	x	CFDRM FDSTsb	RMPTR													—	—	—	—	—	—	—	—	RMIFL	—	—	—	—	—	—	—	—	RMFDF	RMBS	RMESI
R3	0	CFDRM DFbp	RMDB_HH						RMDB_HL						RMDB_LH						RMDB_LL														
R[4...18]	[1...15]	CFDRM DFbp	RMDB_HH						RMDB_HL						RMDB_LH						RMDB_LL														

### 32.2.61.8 CFDRFIDb : RX FIFO Access ID Register b (b = 0 to 1)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0520 + 0x004C × b

Bit position: 31 30 29 0

Bit field:

RFIDE	RFRTTR	—	RFID[28:0]																								
-------	--------	---	------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Value after reset: 0

Bit	Symbol	Function	R/W
28:0	RFID[28:0]	RX FIFO Buffer ID Field STD-ID/EXT-ID fields	R
29	—	This bit is read as 0.	R
30	RFRTTR	RX FIFO Buffer RTR bit 0: Data frame 1: Remote frame	R
31	RFIDE	RX FIFO Buffer IDE bit 0: STD-ID has been received 1: EXT-ID has been received	R

Note: S-TYPE-3, P-TYPE-3

The RX FIFO Access ID Registers b (b = 0 to 1) store the ID field, IDE bit and RTR bit of the message.

#### RFID[28:0] bits (RX FIFO Buffer ID Field)

The RFID[28:0] bits are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.

For alignment of these bits in standard and extended frame format, see Identifier Bits Alignment.

#### RFRTTR bit (RX FIFO Buffer RTR bit)

The RFRTTR bit shows whether a data frame or a remote frame was stored in the FIFO buffer.

Note: There are no remote frames in CANFD format. When a CANFD frame was received, the register reflects the state of the received value (RRS bit in FD frame format).

#### RFIDE bit (RX FIFO Buffer IDE bit)

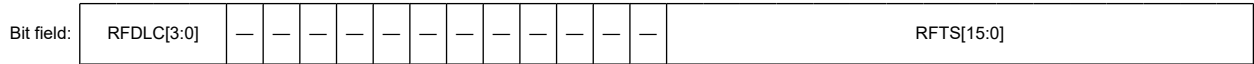
The RFIDE bit shows whether message with the Standard Identifier or Extended Identifier was received in the FIFO buffer.

### 32.2.61.9 CFDRFPTRb : RX FIFO Access Pointer Register b (b = 0 to 1)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0524 + 0x004C × b

Bit position: 31 28 15 0



Value after reset: 0

Bit	Symbol	Function	R/W
15:0	RFTS[15:0]	RX FIFO Timestamp Value Timestamp value of the received CAN frame	R
27:16	—	These bits are read as 0.	R
31:28	RFDLC[3:0]	RX FIFO Buffer DLC Field Number of data bytes received in a CAN frame	R

Note: S-TYPE-3, P-TYPE-3

The FIFO Access Pointer Registers b (b = 0 to 1) store the DLC and Timestamp fields for the received message.

#### RFTS[15:0] bits (RX FIFO Timestamp Value)

The RFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDFGDCFG.TSCCFG bit of the received message.

#### RFDLC[3:0] bits (RX FIFO Buffer DLC Field)

The RFDLC[3:0] bits store the number of data bytes that were received in the RX FIFO buffer.

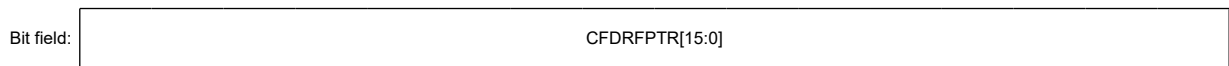
See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

### 32.2.61.10 CFDRFFDSTSb : RX FIFO Access CANFD Status Register b (b = 0 to 1)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

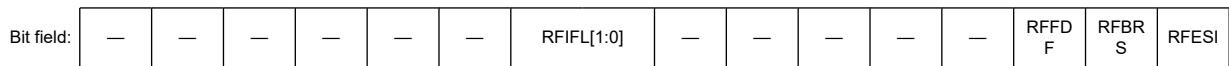
Offset address: 0x0528 + 0x004C × b

Bit position: 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	RFESI*1	Error State Indicator bit 0: CANFD frame received from error active node 1: CANFD frame received from error passive node	R
1	RFBRS*1	Bit Rate Switch bit 0: CANFD frame received with no bit rate switch 1: CANFD frame received with bit rate switch	R
2	RFFDF*1	CAN FD Format bit 0: Non CANFD frame received 1: CANFD frame received	R
7:3	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
9:8	RFIFL[1:0]	RX FIFO Buffer Information Label Field	R
15:10	—	These bits are read as 0.	R
31:16	CFDRFPTR[15:0]	RX FIFO Buffer Pointer Field	R

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is not available in the classical CAN function.

The RX FIFO Access CANFD Status Registers b (b = 0 to 1) show the status of the FDF, BRS, and ESI bits, including the pointer of the received CANFD frame.

**RFESI bit (Error State Indicator bit)**

The RFESI bit has the same value as the ESI bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

**RFBRS bit (Bit Rate Switch bit)**

The RFBRS bit has the same value as the BRS bit of the received CANFD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

**RFFDF bit (CAN FD Format bit)**

The RFFDF bit has the same value as the FDF bit of the received CANFD frame.

Note: This bit is not available in the classical CAN function.

**RFIFL[1:0] bits (RX FIFO Buffer Information Label Field)**

The RFIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

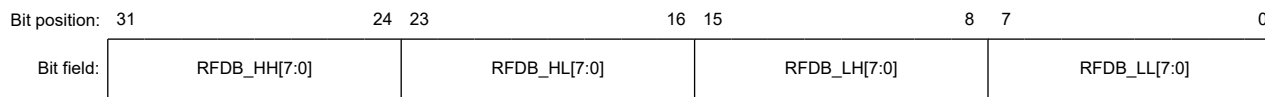
**CFDRFPTR[15:0] bits (RX FIFO Buffer Pointer Field)**

The CFDRFPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

**32.2.61.11 CFDRFDFb<sub>p</sub> : RX FIFO Access Data Field p Register b (p = 0 to 15, b = 0 to 1)**

Base address: CANFD<sub>n</sub> = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFD<sub>n</sub>\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x052C + 0x004 × p + 0x04C × b



Value after reset: 0

Bit	Symbol	Function	R/W
7:0	RFDB_LL[7:0]	RX FIFO Buffer Data Byte (p × 4)	R
15:8	RFDB_LH[7:0]	RX FIFO Buffer Data Byte ((p × 4) + 1)	R
23:16	RFDB_HL[7:0]	RX FIFO Buffer Data Byte ((p × 4) + 2)	R
31:24	RFDB_HH[7:0]	RX FIFO Buffer Data Byte ((p × 4) + 3)	R

Note: S-TYPE-3, P-TYPE-3

The RX FIFO Access Data Field p Registers b (p = 0 to 15, b = 0 to 1) store data bytes ((p × 4) to data byte ((p × 4) + 3) of the received message.

**RFDB\_LL[7:0] bits (RX FIFO Buffer Data Byte ( $p \times 4$ ))**

The RFDB\_LL[7:0] bits store data bytes ( $p \times 4$ ) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00 according to the configured data payload size `CFDRFCCa.RFPLS`.

**RFDB\_LH[7:0] bits (RX FIFO Buffer Data Byte ( $(p \times 4) + 1$ ))**

The RFDB\_LH[7:0] bits store data bytes ( $(p \times 4) + 1$ ) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

**RFDB\_HL[7:0] bits (RX FIFO Buffer Data Byte ( $(p \times 4) + 2$ ))**

The RFDB\_HL[7:0] bits store data bytes ( $(p \times 4) + 2$ ) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

**RFDB\_HH[7:0] bits (RX FIFO Buffer Data Byte ( $(p \times 4) + 3$ ))**

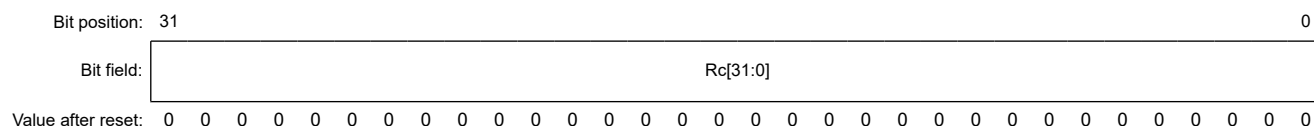
The RFDB\_HH[7:0] bits store data bytes ( $(p \times 4) + 3$ ) of the message present in the FIFO buffer.

Unused data bytes are filled with 0x00.

**32.2.61.12 CFDCFMBCP0[0] : Common FIFO Access Message Buffer Component**

Base address: CANFDn = 0x4038\_0000 + 0x2000  $\times$  n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000  $\times$  n (n = 0, 1)

Offset address: See [Table 32.4](#)



Bit	Symbol	Function	R/W
31:0	Rc[31:0]	Common FIFO Access Message Buffer Component c Refer to <a href="#">Table 32.9</a> , <a href="#">Table 32.10</a> and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R

Note: S-TYPE-3, P-TYPE-3

Where the total number of CFDCFMBCP0 = 1 as shown in [Figure 32.28](#) (c = Common FIFO Message Buffer Component Register index = [0...18])

**Rc[31:0] bit (Common FIFO Access Message Buffer Component c)**

The Common FIFO Access Message Buffer Component is made up of the following registers: CFDCFID, CFDCFPTR, CFFDSTS0, and CFDCDFDp. Refer to [Table 32.10](#) for details of how to interpret the structure of this buffer component and how to access the respective registers.

**Table 32.9 Common FIFO Access Message Buffer Component Summary (1 of 2)**

Common FIFO Access Message Buffer Component (CFMBCP)	
Rc	CANFD mode (CAN_FD_MODE = 1)
R0	Common FIFO Access ID Registers
R1	Common FIFO Access Pointer Register
R2	Common FIFO Access CANFD Status Registers
R3	Common FIFO Access Data Field 0 Registers
R4	Common FIFO Access Data Field 1 Registers
R5	Common FIFO Access Data Field 2 Registers
R6	Common FIFO Access Data Field 3 Registers
R7	Common FIFO Access Data Field 4 Registers



**Table 32.9 Common FIFO Access Message Buffer Component Summary (2 of 2)**

Common FIFO Access Message Buffer Component (CFMBCP)	
Rc	CANFD mode (CAN_FD_MODE = 1)
R8	Common FIFO Access Data Field 5 Registers
R9	Common FIFO Access Data Field 6 Registers
R10	Common FIFO Access Data Field 7 Registers
R11	Common FIFO Access Data Field 8 Registers
R12	Common FIFO Access Data Field 9 Registers
R13	Common FIFO Access Data Field 10 Registers
R14	Common FIFO Access Data Field 11 Registers
R15	Common FIFO Access Data Field 12 Registers
R16	Common FIFO Access Data Field 13 Registers
R17	Common FIFO Access Data Field 14 Registers
R18	Common FIFO Access Data Field 15 Registers
R[19...31]	—

**Table 32.10 Common FIFO Access Message Buffer Component (CFMBCP) Detailed**

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	x	CFDCFI D	CFIDE	CFRTR	THLEN	CFID																												
R1	x	CFDCFPTR	CFDLC			—	—	—	—	—	—	—	—	—	—	—	—	—	CFTS															
R2	x	CFDCFFDCSTS	CFPTR															—	—	—	—	—	—	CFIFL	—	—	—	—	—	—	CFDF	CFBRS	CFESI	
R3	0	CFDCDFp	CFDB_HH					CFDB_HL					CFDB_LH					CFDB_LL																
R[4...18]	[1...15]	CFDCDFp	CFDB_HH					CFDB_HL					CFDB_LH					CFDB_LL																
R[19...31]	x	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	

**32.2.61.13 CFDCFIID : Common FIFO Access ID Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x05B8

Bit position: 31 30 29 28 0



Value after reset: 0

Bit	Symbol	Function	R/W
28:0	CFID[28:0]	Common FIFO Buffer ID Field STD-ID / EXT-ID fields	R/W

Bit	Symbol	Function	R/W
29	THLEN	THL Entry enable TX FIFO Mode: 0: Entry will not be stored in THL after successful TX. 1: Entry will be stored in THL after successful TX. RX FIFO Mode: Reserved, this bit is read as 0	R/W
30	CFRTR	Common FIFO Buffer RTR Bit 0: Data Frame 1: Remote Frame	R/W
31	CFIDE	Common FIFO Buffer IDE Bit 0: STD-ID will be transmitted or has been received 1: EXT-ID will be transmitted or has been received	R/W

Note: S-TYPE-3, P-TYPE-3

The Common FIFO Access ID registers store the ID field, IDE bit and RTR bit of the message.

In TX mode, users can read data from the FIFO, only for the current entry based on the write pointer value, not for the other entries.

**CFID[28:0] bit (Common FIFO Buffer ID Field)**

These are the bits of the STD-ID / EXT-ID fields of the message in the FIFO Buffer.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

**THLEN bit (THL Entry enable)**

This bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

**CFRTR bit (Common FIFO Buffer RTR Bit)**

This bit selects whether a Data Frame or a Remote Frame will be transmitted from or was received in the FIFO Buffer.

Note: There are no remote frames in CANFD format. In case a CANFD frame was received (RX mode) the register reflects the state of the received value (RRS bit in FD frame format). In case of CANFD transmission (TX mode CFDCFID.CFFDF = 1) the bit is always transmitted dominant (Data Frame).

In TX mode, users can write and read from FIFO buffers.

In RX mode, users can only read data from FIFO buffers.

**CFIDE bit (Common FIFO Buffer IDE Bit)**

This bit selects whether a message with EXT-ID or STD-ID will be transmitted from or was received in the FIFO Buffer.

In TX mode, users can write and read from FIFO buffers.

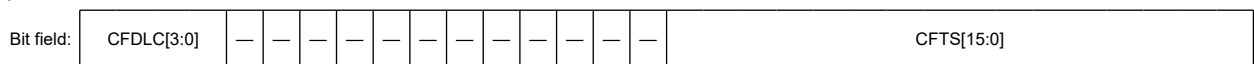
In RX mode, users can only read data from FIFO buffers.

**32.2.61.14 CFDCFPTR : Common FIFO Access Pointer Register**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x05BC

Bit position: 31 28 15 0



Value after reset: 0

Bit	Symbol	Function	R/W
15:0	CFTS[15:0]	Common FIFO Timestamp Value Timestamp value of the received CAN frame (FIFO in RX mode).	R/W
27:16	—	These bits are read as 0. The write value should be 0.	R/W
31:28	CFDLC[3:0]	Common FIFO Buffer DLC Field Number of data bytes received in a CAN frame, or to be transmitted in a CAN frame.	R/W

Note: S-TYPE-3, P-TYPE-3

The Common FIFO Access Pointer Registers store the DLC and Timestamp fields.

In TX mode, you can read data from the FIFO buffer, only for the current entry based on the write pointer value, and not for the other entries.

### CFTS[15:0] bits (Common FIFO Timestamp Value)

The CFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message (if FIFO is configured in RX mode).

In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

### CFDLC[3:0] bits (Common FIFO Buffer DLC Field)

The CFDLC[3:0] bits store the number of data bytes that were received in the FIFO buffer or are to be transmitted.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes.

In TX mode, you can read and write from the FIFO buffers. Do not read data for the other entries in the FIFO when configured in TX mode.

In RX mode, you can only read data from the FIFO buffers.

## 32.2.61.15 CFDCFFDCSTS : Common FIFO Access CANFD Control/Status Register

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x05C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CFIFL[1:0]	—	—	—	—	—	—	CFDF	CFBR	CFESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFESI*1	Error State Indicator bit 0: CANFD frame received or to transmit by error active node 1: CANFD frame received or to transmit by error passive node	R/W
1	CFBRS*1	Bit Rate Switch bit 0: CANFD frame received or to transmit with no bit rate switch 1: CANFD frame received or to transmit with bit rate switch	R/W
2	CFDF*1	CAN FD Format bit 0: Non CANFD frame received or to transmit 1: CANFD frame received or to transmit	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CFIFL[1:0]	COMMON FIFO Buffer Information Label Field	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31:16	CFPTR[15:0]	Common FIFO Buffer Pointer Field	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is not available in the classical CAN function.

The Common FIFO Access CANFD Control/Status Registers show the status of the FDF, BRS and ESI bits, including the pointer of the received CANFD frame or the CANFD frame to transmit.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

#### CFESI bit (Error State Indicator bit)

In TX mode, you can read and write from FIFO buffers. In this mode, when the CANFD module is not in error passive, the CFESI bit equals the write value. Otherwise, it is a don't care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

In RX mode, you can only read data from FIFO buffers.

In RX mode, the CFESI bit is updated with the ESI bit value of the CANFD frame when it has been received, indicating the error state of the transmitting node. In RX mode, 0 is stored to this bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

#### CFBRS bit (Bit Rate Switch bit)

In TX mode, you can read and write from FIFO buffers. In this mode, the CANFD module either transmits a 0 to indicate no bit rate switch in the frame to be transmitted or a 1 to indicate a bit rate switch in the frame to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In RX mode, the CFBRS bit is updated with the BRS bit value of the CANFD frame when it has been received, indicating whether there is a bit rate switch (1) or (0) on the CANFD frame.

In RX mode, 0 is stored to the CFBRS bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

#### CFFDF bit (CAN FD Format bit)

In TX mode, you can read and write from FIFO buffers. In this mode, the CANFD module either transmits a 0 to indicate a CAN 2.0 frame is to be transmitted or a 1 to indicate a CANFD frame is to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In RX mode, the CFFDF bit is updated with the FDF bit value of the CAN frame when it has been received, indicating whether it is a CAN 2.0 frame (0) or a CANFD frame (1).

Note: This bit is not available in the classical CAN function.

#### CFIFL[1:0] bits (COMMON FIFO Buffer Information Label Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTS.CFIFL[1:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The information label value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX mode).

In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

#### CFPTR[15:0] bits (Common FIFO Buffer Pointer Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTS.CFPTR[15:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The pointer value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX mode).

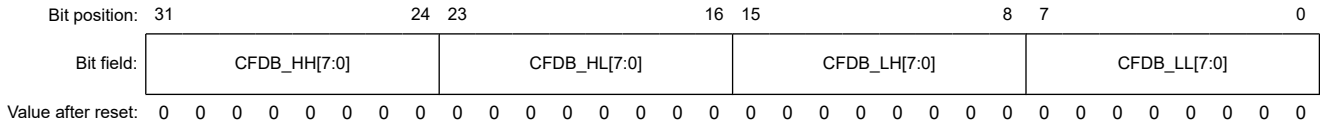
In TX mode, you can read and write from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

### 32.2.61.16 CFDCFDp : Common FIFO Access Data Field p Registers (p = 0 to 15)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x05C4 + 0x004 × p



Bit	Symbol	Function	R/W
7:0	CFDB_LL[7:0]	Common FIFO Buffer Data Bytes (p × 4)	R/W
15:8	CFDB_LH[7:0]	Common FIFO Buffer Data Bytes ((p × 4) + 1)	R/W
23:16	CFDB_HL[7:0]	Common FIFO Buffer Data Bytes ((p × 4) + 2)	R/W
31:24	CFDB_HH[7:0]	Common FIFO Buffer Data Bytes ((p × 4) + 3)	R/W

Note: S-TYPE-3, P-TYPE-3

The FIFO Access Data Field p Registers (p = 0 to 15) store data bytes (p × 4) to data bytes ((p × 4) + 3) of the message.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

#### CFDB\_LL[7:0] bits (Common FIFO Buffer Data Bytes (p × 4))

The CFDB\_LL[7:0] bits store data bytes (p × 4) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.\*1

#### CFDB\_LH[7:0] bits (Common FIFO Buffer Data Bytes ((p × 4) + 1))

The CFDB\_LH[7:0] bits store data bytes ((p × 4) + 1) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.\*1

#### CFDB\_HL[7:0] bits (Common FIFO Buffer Data Bytes ((p × 4) + 2))

The CFDB\_HL[7:0] bits store data bytes ((p × 4) + 2) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.\*1

#### CFDB\_HH[7:0] bits (Common FIFO Buffer Data Bytes ((p × 4) + 3))

The CFDB\_HH[7:0] bits store data bytes ((p × 4) + 3) of the message present in the FIFO buffer.

In TX mode, you can read and write from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In RX mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCC.CFPLS.\*1

Note 1. In RX mode, unused data bytes are filled with 0x00 according to the configured data payload size CFDCFCC.CFPLS, which is a CANFD feature not found in classical CAN.

### 32.2.61.17 CFDTMBCPb[0] : TX Message Buffer Component b (b = 0 to 3)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: See [Table 32.4](#)

Bit position: 31

0

Bit field:

Rc[31:0]

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	Rc[31:0]	TX Message Buffer Component c Refer to <a href="#">Table 32.11</a> , <a href="#">Table 32.12</a> and the descriptions that follow for a detailed description of each register and its related bits, contained within this message buffer component.	R

Note: S-TYPE-3, P-TYPE-3

Where the total number of CFDTMBCPn = 4 as shown in [Figure 32.28](#) (c = TX Message Buffer Component Register index = [0...18])

#### Rc[31:0] bit (TX Message Buffer Component c)

TX Message Buffer Component c

The TX Message Buffer Component is made up of the following registers: CFDTMIDb, CFDTMPTRb, CFDTMFDCTRb, and CFDTMDFbp. Refer to [Table 32.12](#) for details of how to interpret the structure of this buffer component and how to access the respective registers.

**Table 32.11 TX Message Buffer Component Summary**

TX Message Buffer Component (TMBCP)	
Rc	CANFD mode (CAN_FD_MODE = 1b)
R0	TX Message Buffer (b) ID Registers
R1	TX Message Buffer (b) Pointer Registers
R2	TX Message Buffer (b) CANFD Status Registers
R3	TX Message Buffer (b) Data Field 0 Registers
R4	TX Message Buffer (b) Data Field 1 Registers
R5	TX Message Buffer (b) Data Field 2 Registers
R6	TX Message Buffer (b) Data Field 3 Registers
R7	TX Message Buffer (b) Data Field 4 Registers
R8	TX Message Buffer (b) Data Field 5 Registers
R9	TX Message Buffer (b) Data Field 6 Registers
R10	TX Message Buffer (b) Data Field 7 Registers
R11	TX Message Buffer (b) Data Field 8 Registers
R12	TX Message Buffer (b) Data Field 9 Registers
R13	TX Message Buffer (b) Data Field 10 Registers
R14	TX Message Buffer (b) Data Field 11 Registers
R15	TX Message Buffer (b) Data Field 12 Registers
R16	TX Message Buffer (b) Data Field 13 Registers
R17	TX Message Buffer (b) Data Field 14 Registers
R18	TX Message Buffer (b) Data Field 15 Registers
R[19...31]	—

**Table 32.12 TX Message Buffer Component (TMBCP) Detailed**

Rc	p	Symbol	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0					
R0	x	CFDTMI Db	TMIDE	TMRTR	THLEN	TMID																																	
R1	x	CFDTM PTRb	TMDLC			—	—	—	—	—	—	—	—	—	—	—	—	—	CFTS																				
R2	x	CFDTM FDCTRb	TMPTR															—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
R3	0	CFDTM DFbp	TMDB_HH					TMDB_HL					TMDB_LH					TMDB_LL																					
R[4...18]	[1...15]	CFDTM DFbp	TMDB_HH					TMDB_HL					TMDB_LH					TMDB_LL																					

**32.2.61.18 CFDTMIDb : TX Message Buffer ID Registers (b = 0 to 3)**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0604 + 0x004C × b

Bit position: 31 30 29 0



Value after reset: 0

Bit	Symbol	Function	R/W
28:0	TMID[28:0]	TX Message Buffer ID Field STD-ID/EXT-ID fields	R/W
29	THLEN	Tx History List Entry 0: Entry not stored in THL after successful TX 1: Entry stored in THL after successful TX	R/W
30	TMRTR	TX Message Buffer RTR bit 0: Data frame 1: Remote frame	R/W
31	TMIDE	TX Message Buffer IDE bit 0: STD-ID is transmitted 1: EXT-ID is transmitted	R/W

Note: S-TYPE-3, P-TYPE-3

Each TX Message Buffer ID Register b (b = 0 to 3) are used to store the ID, IDE, RTR fields and history configuration of the message to be transmitted from the associated buffer.

**TMID[28:0] bits (TX Message Buffer ID Field)**

The TMID[28:0] bits are bits of the STD-ID/EXT-ID fields of the message stored in this TX message buffer.

Do not write to these bits when the related CANFD channel is in CH\_SLEEP mode.

**THLEN bit (Tx History List Entry)**

The THLEN bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

Do not write to these bits when the related CANFD channel is in CH\_SLEEP mode.

**TMRTR bit (TX Message Buffer RTR bit)**

The TMRTR bit selects whether a data frame or remote frame is to be transmitted from this TX message buffer.

Note: There are no remote frames in CANFD format. For a CANFD transmission (CFDTRMFDCTRb.CFFDF = 1), this bit is always transmitted dominant (data frame).

Do not write to these bits when the related CANFD channel is in CH\_SLEEP mode.

**TMIDE bit (TX Message Buffer IDE bit)**

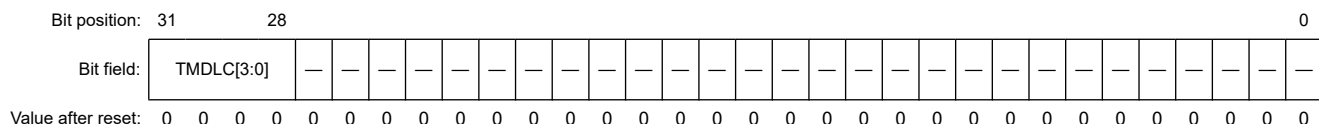
The TMIDE bit selects whether a message with EXT-ID or STD-ID is to be transmitted from this TX message buffer.

Do not write to these bits when the related CANFD channel is in CH\_SLEEP mode.

**32.2.61.19 CFDTMPTRb : TX Message Buffer Pointer Register (b = 0 to 3)**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0608 + 0x004C × b



Bit	Symbol	Function	R/W
27:0	—	The read values are undefined. The write value should be 0.	R/W
31:28	TMDLC[3:0]	TX Message Buffer DLC Field Number of data bytes to be transmitted in a CAN frame.	R/W

Note: S-TYPE-3, P-TYPE-3

Each TX Message Buffer Pointer Register b (b = 0 to 3) is used to store the DLC fields of the message to transmit from the associated buffer.

**TMDLC[3:0] bits (TX Message Buffer DLC Field)**

The TMDLC[3:0] bits select the number of data bytes to be transmitted from this TX message buffer when the corresponding TMRTR bit is configured as 0.

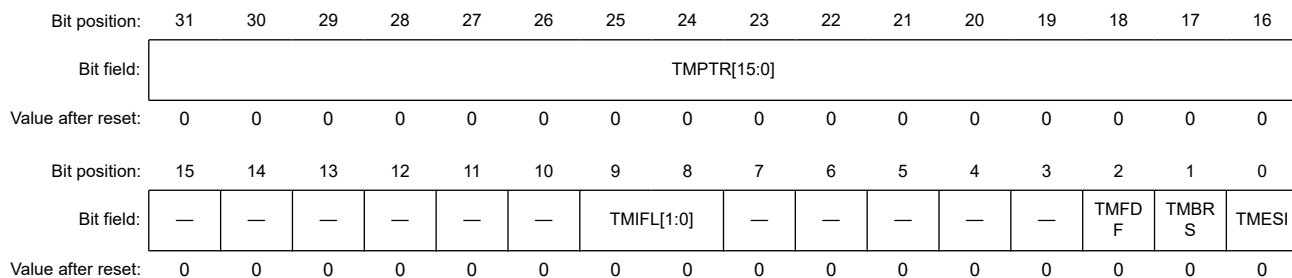
See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes to be transmitted.

Do not write to these bits when the related CANFD channel is in CH\_SLEEP mode.

**32.2.61.20 CFDTMFDCTRb : TX Message Buffer CANFD Control Register (b = 0 to 3)**

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x060C + 0x004C × b





Bit	Symbol	Function	R/W
0	TMESI*1	Error State Indicator bit 0: CANFD frame to transmit by error active node 1: CANFD frame to transmit by error passive node	R/W
1	TMBRS*1	Bit Rate Switch bit 0: CANFD frame to transmit with no bit rate switch 1: CANFD frame to transmit with bit rate switch	R/W
2	TMFDF*1	CAN FD Format bit 0: Non CANFD frame to transmit 1: CANFD frame to transmit	R/W
7:3	—	The read values are undefined. The write value should be 0.	R/W
9:8	TMIFL[1:0]	TX Message Buffer Information Label Field	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W
31:16	TMPTR[15:0]	TX Message Buffer Pointer Field	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is not available in the classical CAN function.

The TX Message Buffer CANFD Control Registers b (b = 0 to 3) show the status of the FDF, BRS and ESI bits, including the pointer fields of the CANFD frame to be transmitted.

#### TMESI bit (Error State Indicator bit)

If the channel is not in error passive, then the TMESI bit equals the write value, otherwise it is a don't care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

Do not write to the TMESI bit when the related CANFD channel is in CH\_SLEEP mode.

Note: This bit is not available in the classical CAN function.

#### TMBRS bit (Bit Rate Switch bit)

Do not write to the TMBRS bit when the related CANFD channel is in CH\_SLEEP mode.

Note: This bit is not available in the classical CAN function.

#### TMFDF bit (CAN FD Format bit)

Do not write to the TMFDF bit when the related CANFD channel is in CH\_SLEEP mode.

Note: This bit is not available in the classical CAN function.

#### TMIFL[1:0] bits (TX Message Buffer Information Label Field)

The TMIFL[1:0] bits store the information label value to be copied, together with additional message information, in the TX History List after successful transmission of the message.

Do not write to these bits when the related CANFD channel is in CH\_SLEEP mode.

#### TMPTR[15:0] bits (TX Message Buffer Pointer Field)

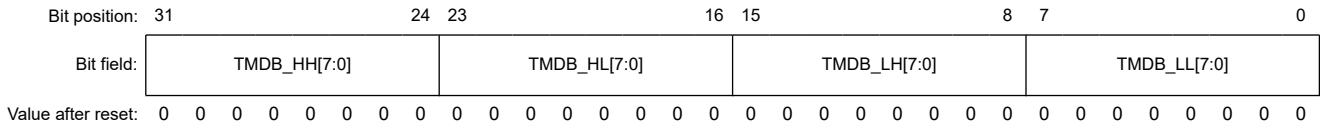
The TMPTR[15:0] bits store the pointer value to be copied, together with additional message information in the TX History List after successful transmission of the message.

Do not write to these bits when the related CANFD channel is in CH\_SLEEP mode.

### 32.2.61.21 CFDTMDFb\_p : TX Message Buffer Data Field Register (p= 0 to 15 , b= 0 to 3)

Base address: CANFDn = 0x4038\_0000 + 0x2000 × n (n = 0, 1)  
 CANFDn\_NS = 0x5038\_0000 + 0x2000 × n (n = 0, 1)

Offset address: 0x0610 + 0x004 × p + 0x004C × b



Bit	Symbol	Function	R/W
7:0	TMDB_LL[7:0]	TX Message Buffer Data Byte (p × 4)	R/W
15:8	TMDB_LH[7:0]	TX Message Buffer Data Byte ((p × 4) + 1)	R/W
23:16	TMDB_HL[7:0]	TX Message Buffer Data Byte ((p × 4) + 2)	R/W
31:24	TMDB_HH[7:0]	TX Message Buffer Data Byte ((p × 4) + 3)	R/W

Note: S-TYPE-3, P-TYPE-3

Each TX Message Buffer Data Field p Register b (p = 0 to 15, b = 0 to 3) is used to store data bytes (p × 4) to data bytes ((p × 4) + 3) of the message to transmit from the associated buffer.

#### TMDB\_LL[7:0] bits (TX Message Buffer Data Byte (p × 4))

TMDB\_LL[7:0] bits store data bytes (p × 4) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH\_SLEEP mode.

#### TMDB\_LH[7:0] bits (TX Message Buffer Data Byte ((p × 4) + 1))

TMDB\_LH[7:0] bits store data bytes ((p × 4) + 1) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH\_SLEEP mode.

#### TMDB\_HL[7:0] bits (TX Message Buffer Data Byte ((p × 4) + 2))

TMDB\_HL[7:0] bits store data bytes ((p × 4) + 2) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH\_SLEEP mode.

#### TMDB\_HH[7:0] bits (TX Message Buffer Data Byte ((p × 4) + 3))

TMDB\_HH[7:0] bits store data bytes ((p × 4) + 3) of the message in the TX message buffer.

Do not write to these bits when the related CANFD channel is in CH\_SLEEP mode.

## 32.3 Modes of Operation

### 32.3.1 Overview

The modes of the CANFD module can be classified into 2 groups:

- Global modes
- Channel modes

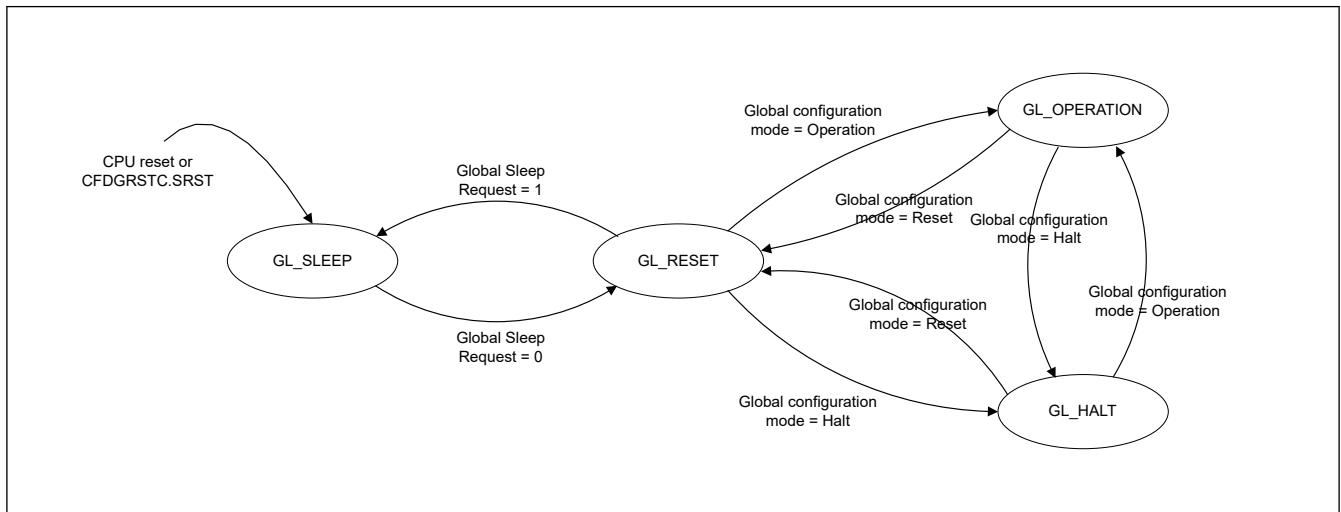
### 32.3.2 Global Modes

These modes are applicable for the complete CANFD module and therefore are called Global modes. The global modes of the CANFD module are:

- Global Sleep
- Global Reset
- Global Halt

- Global Operation.

Figure 32.2 shows the possible transitions between the Global modes.



**Figure 32.2 Transition between CANFD Global modes**

Change in the Global mode can affect the Channel mode. Table 32.13 shows the effect of a Global mode transition on a Channel mode.

**Table 32.13 Possible CANFD Channel modes and Global modes**

Current Global mode	Target Global mode			
	Sleep	Reset	Halt	Operation
<b>Sleep</b>		Ch-Sleep: Keep Ch-Reset: N/A Ch-Halt: N/A Ch-Oper: N/A		
<b>Reset</b>	Ch-Sleep: Keep Ch-Reset: → Ch-Sleep Ch-Halt: N/A Ch-Oper: N/A		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A
<b>Halt</b>		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: N/A		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: N/A
<b>Operation</b>		Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: → Ch-Reset	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: → Ch-Halt	

### 32.3.2.1 Global Sleep Mode

After the release of a hardware reset or after setting and clearing a CFDGRSTC.SRST bit, the CANFD module automatically enters Global Sleep mode.

The CANFD module also enters the Global Sleep mode when the Global Sleep Request bit is set while it is in Global Reset mode. This control bit cannot be set in Global Halt mode or Global Operation mode.

Setting the Global Sleep Request bit sets Channel Sleep Request bit and forces the channel into the Channel Sleep mode.

Sleep mode is used for power saving purpose. When CANFD module is in Global Sleep mode, only the clock for CPU write access to the Global Sleep Mode Request bit is active. All other clocks are stopped and all other functions of the CANFD module are suspended.

Read access from all registers is still possible and all register values are preserved.

After setting the Global Sleep Request bit, it is necessary to confirm that the Global Sleep status has been updated, indicating successful transition to Global Sleep mode before the Global Sleep Request bit can be cleared again.

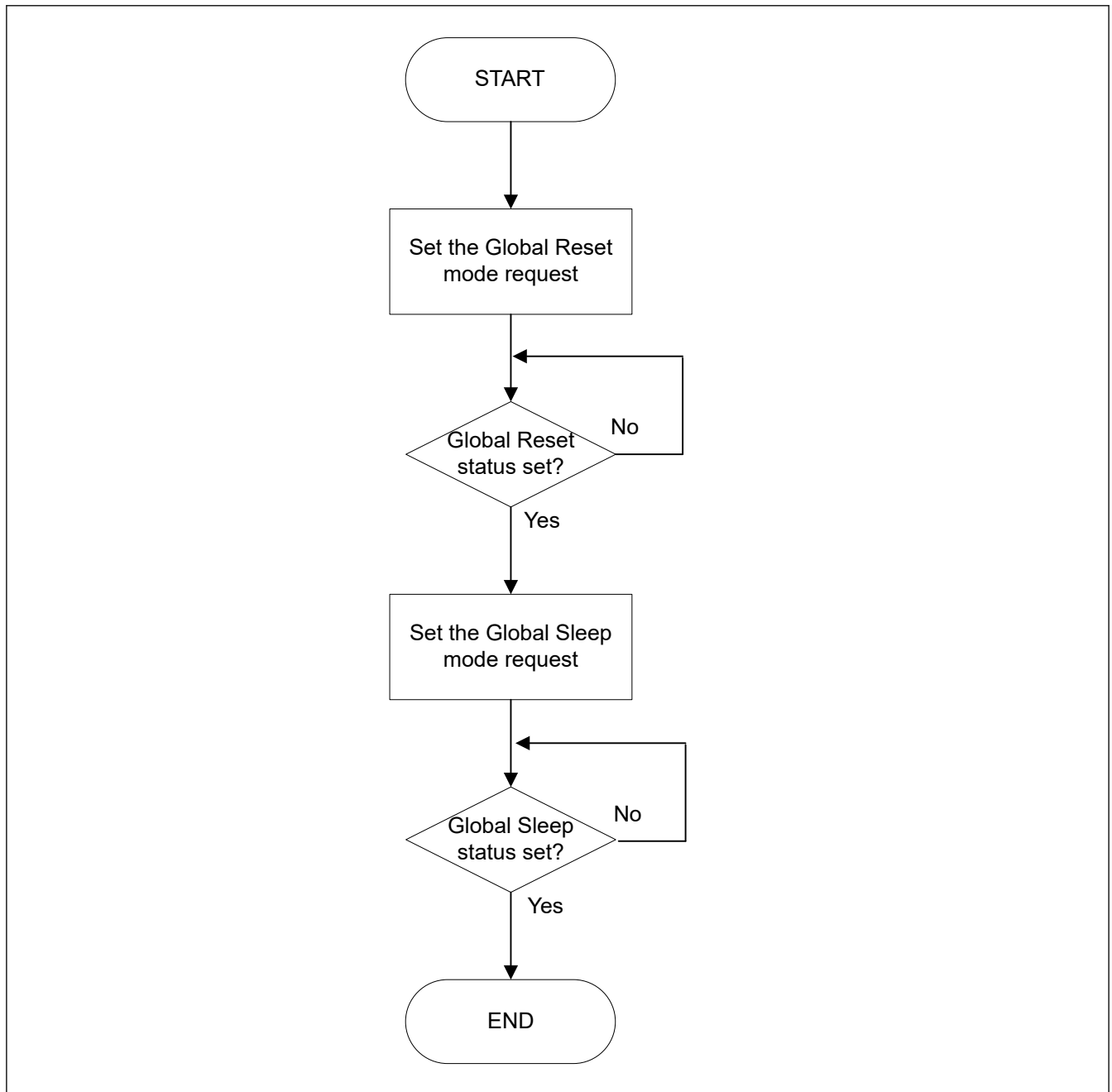
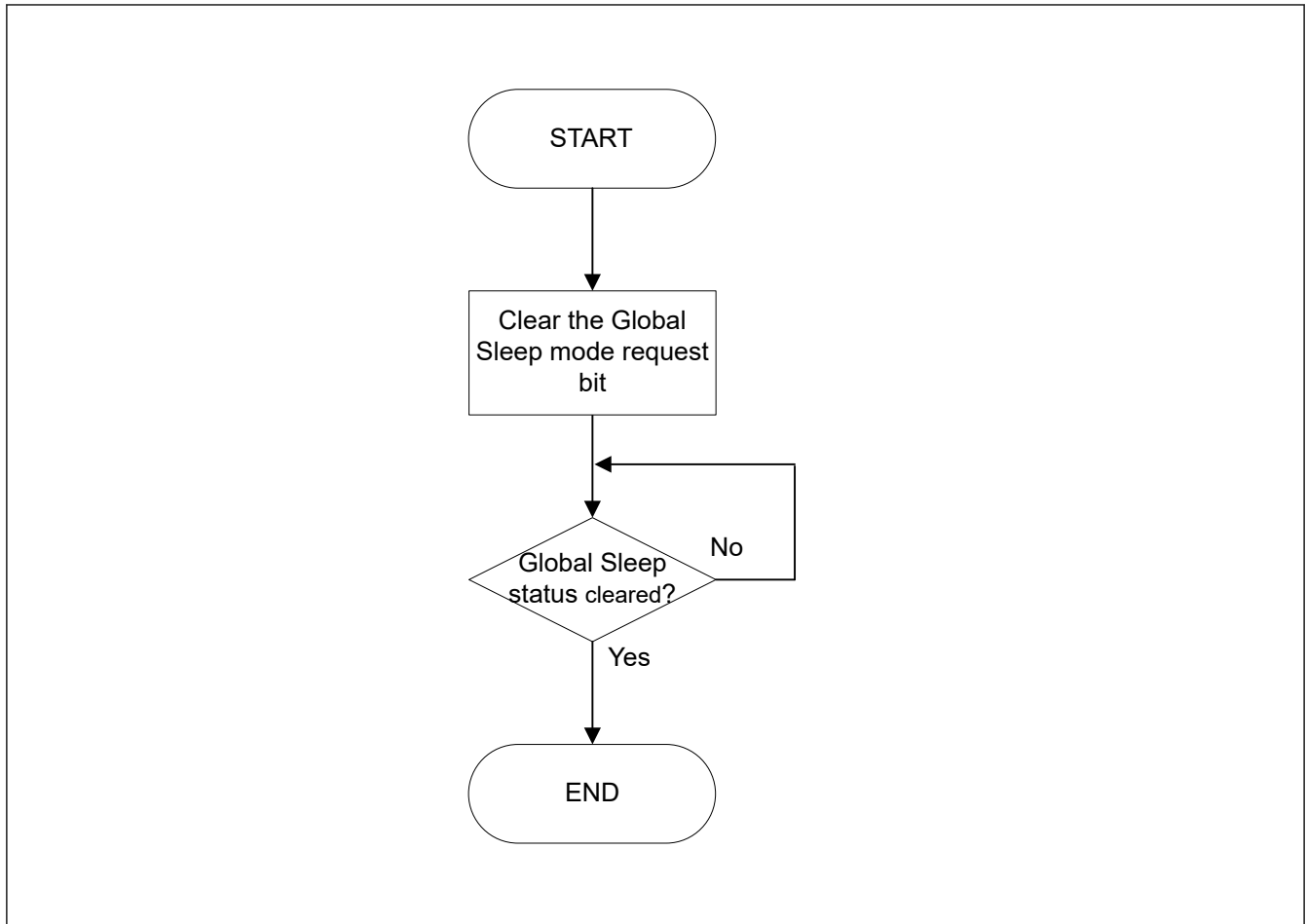


Figure 32.3 Procedure for entering Global Sleep mode



**Figure 32.4** Procedure for exiting Global Sleep mode

### 32.3.2.2 Global Reset Mode

The CANFD module enters this mode in the following ways:

- Global Mode Control bit `CFDGCTR.GMDC` in the Global Control Register is configured for Global Reset mode while the CANFD module is in Global Halt or Global Operation mode
- Global Sleep Mode Request bit is cleared while CANFD module is in Global Sleep mode.

In Global Reset mode, all CANFD module functions are suspended and all status and flag registers are initialized.

Additionally all FIFOs and TX Queues are disabled and transmission control bits are cleared.

Configuration registers (except the test mode registers) are not initialized in this mode to their MCU reset values and the CANFD module can be configured.

See [section 32.3.4. Global Mode and Channel Mode Transition Interactions](#) for a detailed description of the behavior of all registers when transition to Global Reset mode is performed.

Setting the Global mode to Reset by setting the Global Mode Control bits `CFDGCTR.GMDC` in the Global Control Register to 01b sets Channel Mode Control bits `CFDC0CTR.CHMDC` in the Channel Control Registers to 01b and forces the channel into the Channel Reset mode.

For channels that are already in Channel Reset mode or Channel Sleep mode, this automatic transition is not performed (`CFDC0CTR.CHMDC` of related channel already set to 01b).

After setting Global Mode Control bit `CFDGCTR.GMDC` to Reset mode, it is necessary to confirm that the Reset Mode Status bit `CFDGSTS.GRSTSTS` in the Global Status Register has been updated, indicating successful transition to Global Reset mode before `CFDGCTR.GMDC` can be changed again.

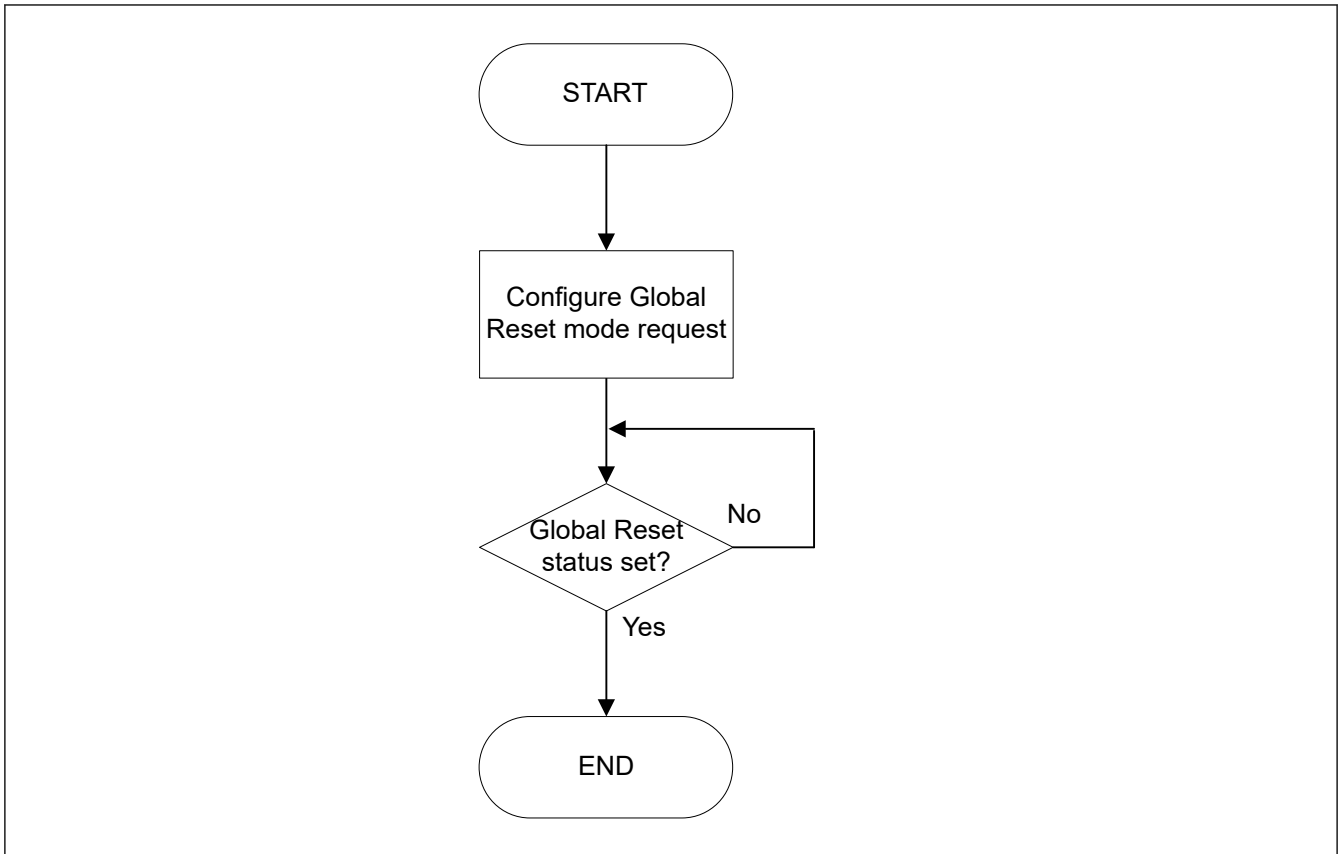
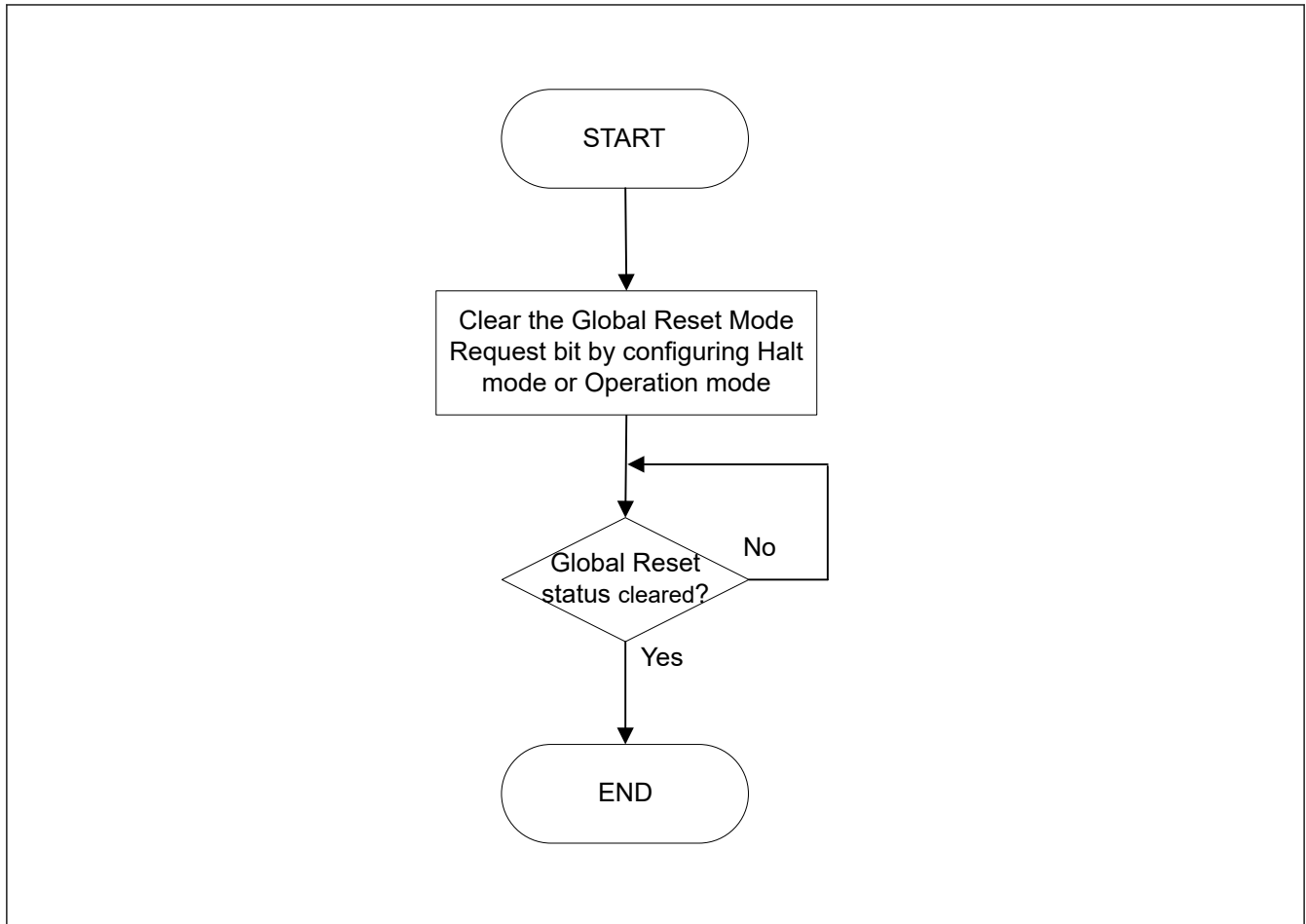


Figure 32.5 Procedure for entering Global Reset mode



**Figure 32.6 Procedure for exiting Global Reset mode**

### 32.3.2.3 Global Halt Mode

The CANFD module enters this mode in the following ways:

- Global Mode Control bit `CFDGCTR.GMDC` in the Global Control Register is configured for Global Halt mode while the CANFD module is in Global Reset mode:
  - the channel in either Channel Reset or Channel Sleep mode remains in this mode
- Global Mode Control bit `CFDGCTR.GMDC` in the Global Control Register is configured for Global Halt mode while the CANFD module is in Global Operation mode:
  - the channel in Channel Reset, Channel Halt, or Channel Sleep mode remains in this mode
  - the channel in Channel Operation mode transitions to Channel Halt mode
  - Global Halt Mode Status bit is set when the channel has left Channel Operation mode.

If a transmission or reception is ongoing for a channel, the transition to Channel Halt mode is delayed until completion of the communication.

Similarly, if a channel is in bus-off, the full bus-off recovery sequence may be delayed depending on the channel configuration.

In Global Halt mode, all communications are suspended and CANFD logic does not cause any change to the Status and Flag registers (only when a channel is in the bus-off that its REC and TEC values are cleared). Additionally, the test mode configuration and control registers are not initialized in this mode.

The Global Halt mode should be used to configure global module test modes.

See [section 32.3.4. Global Mode and Channel Mode Transition Interactions](#) for a detailed description of the behavior of all registers when transition to Global Halt mode is performed.

Setting the Global mode to Halt by setting the Global Mode Control bit `CFDGCTR.GMDC` in the Global Control Register to 10b sets Channel Mode Control bits `CFDC0CTR.CHMDC` in the Channel Control Registers to 10b for the channel that are in Channel Operation mode and forces these channels into the Channel Halt mode.

For the channel that are already in Channel Reset, Channel Halt, or Channel Sleep mode, this automatic transition is not performed.

Therefore, the Global Halt mode request can be used to shut down all CANFD channel communications without loss of messages and disruption on the related CAN bus (no interruption of reception/transmission processes on the channels).

After setting the Global Mode Control bit `CFDGCTR.GMDC` to Halt mode, it is necessary to confirm that the Halt Mode Status bit `CFDGSTS.GHLTSTS` in the Global Status Register has been updated to indicate a successful transition to Global Halt mode. Do not specify any other SFR setting until confirming `CFDGSTS.GHLTSTS` is set.

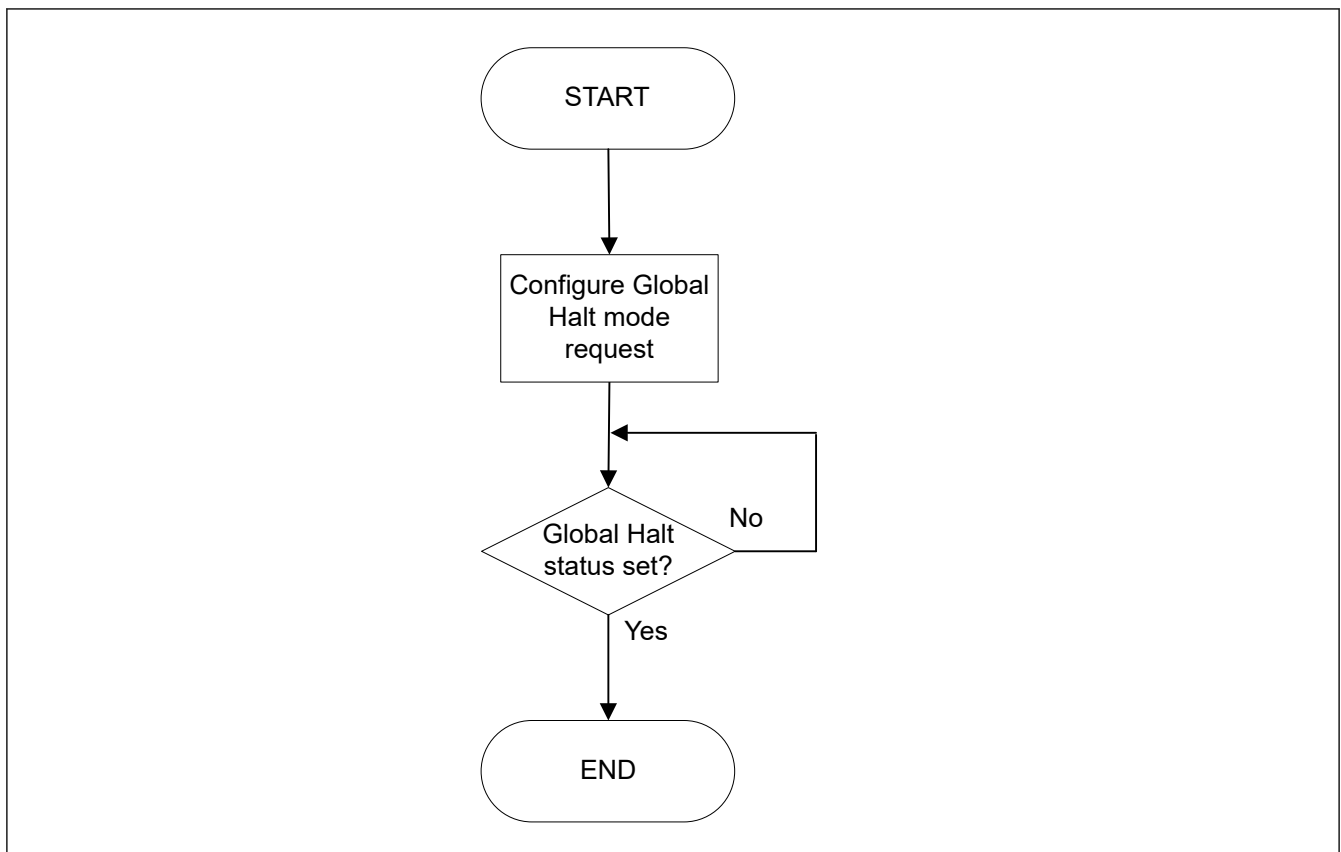
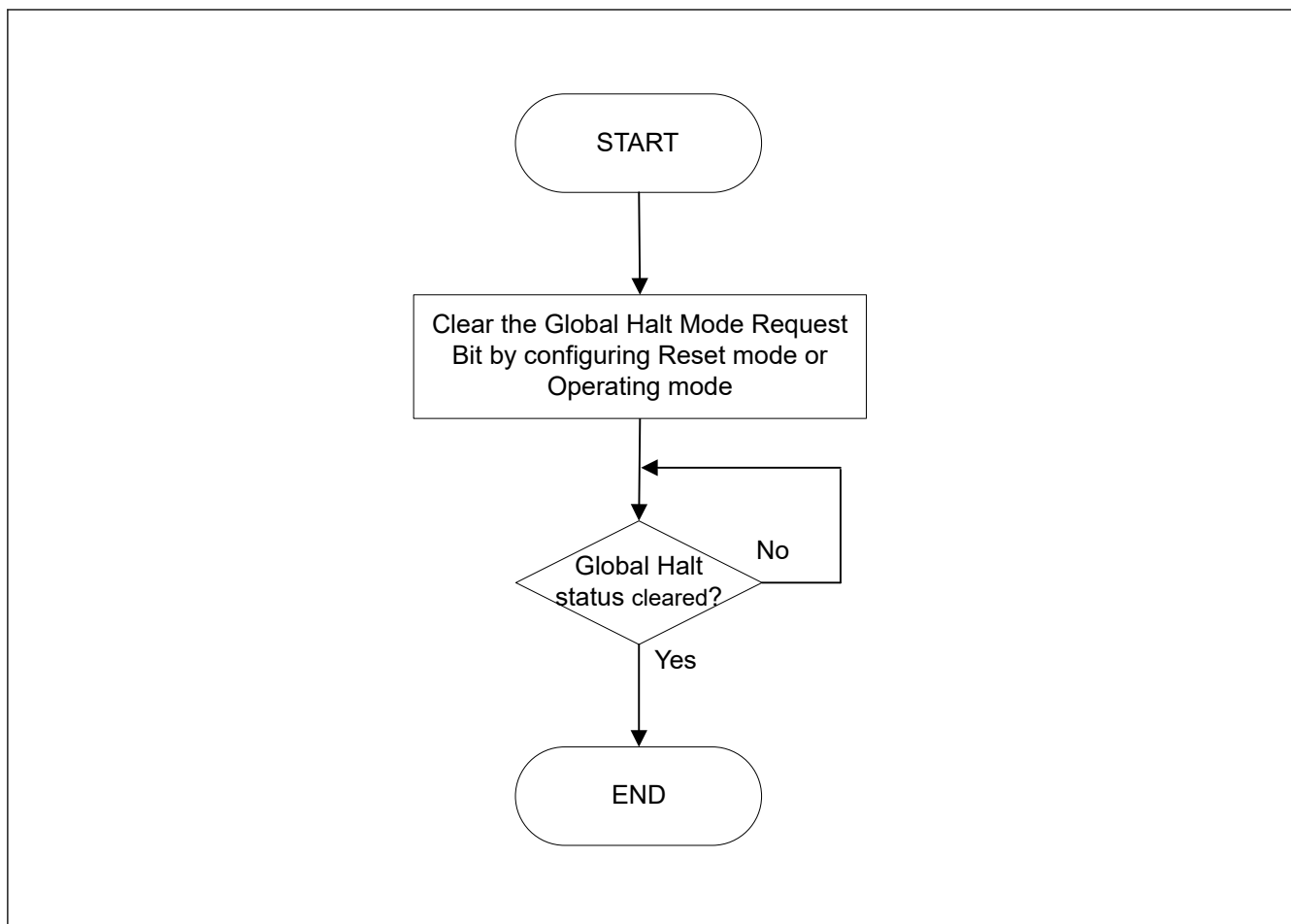


Figure 32.7 Procedure for entering Global Halt mode





**Figure 32.8 Procedure for exiting Global Halt mode**

### 32.3.2.4 Global Operation Mode

The CANFD module enters this mode when the Global Mode Configuration bits are set to Global Operation mode.

The CANFD channel can only be set to Channel Operation mode and start CAN communication when CANFD is in Global Operation mode.

After setting the Global Mode Control bit `CFDGCTR.GMDC` to Global Operation mode, it is necessary to confirm that the Global Reset Mode Status bit `CFDGSTS.GRSTSTS` and the Global Halt Mode Status bit `CFDGSTS.GHLTSTS` in the Global Status Register have been cleared to indicate a successful transition to Global Operation mode before `CFDGCTR.GMDC` can be modified again.

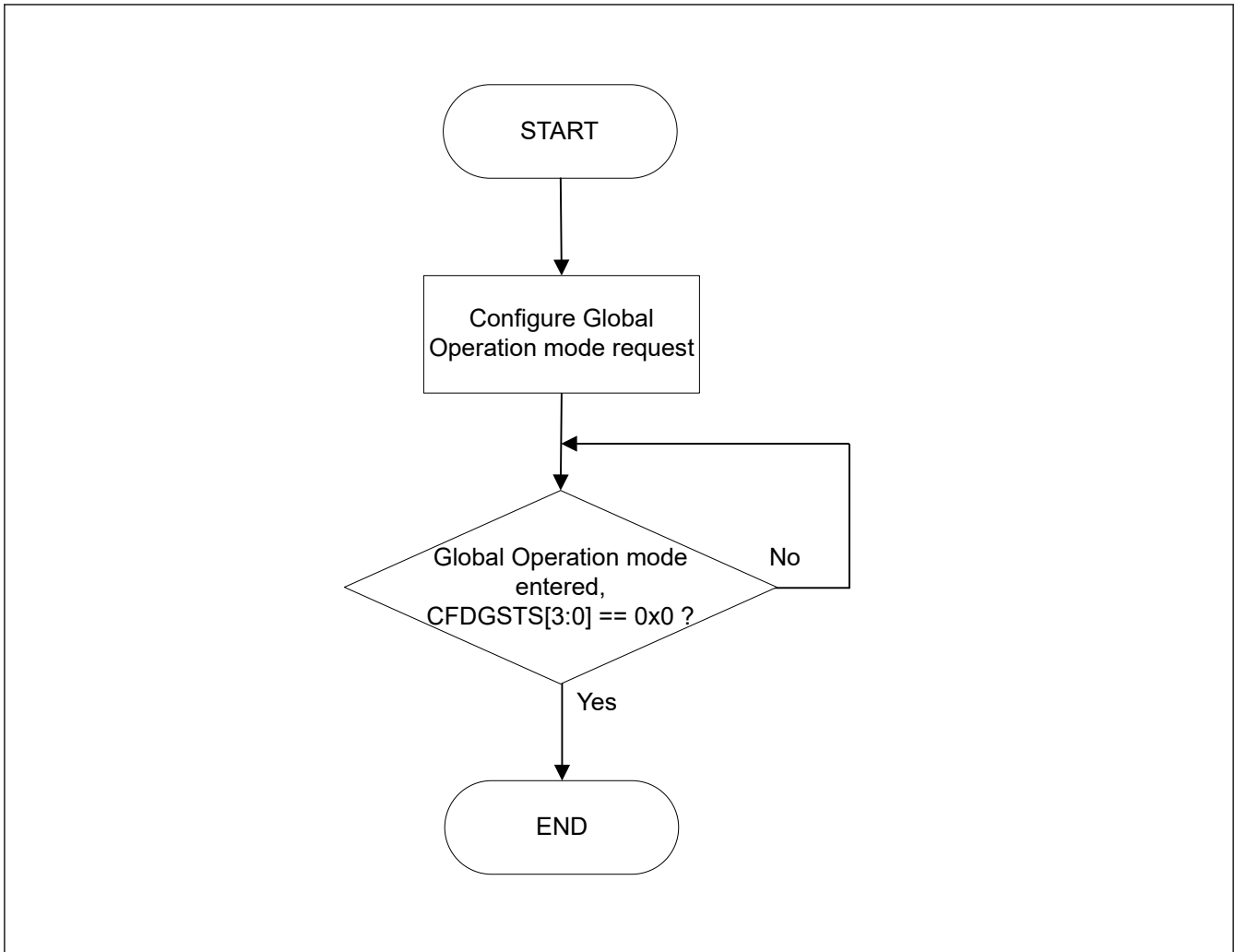
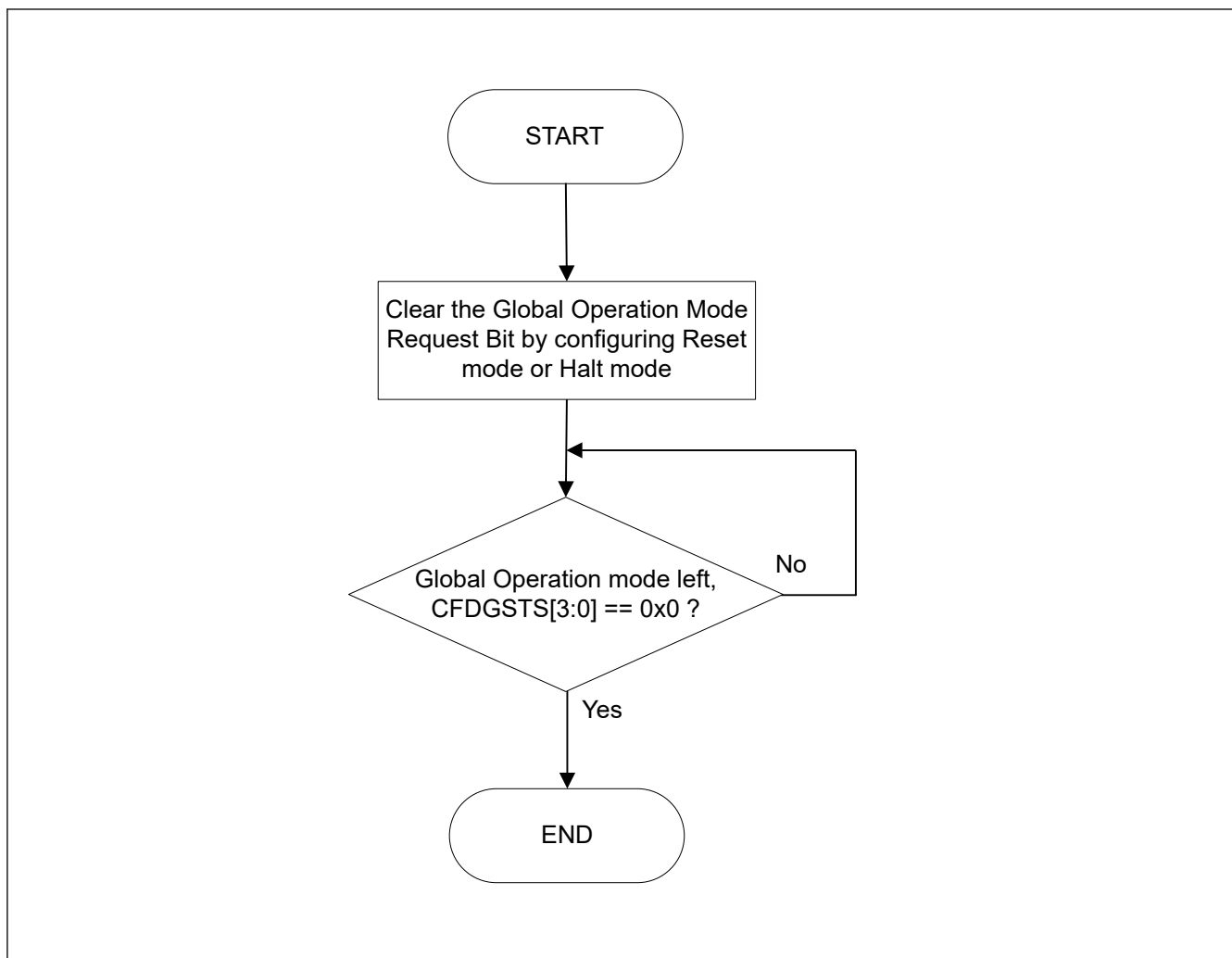


Figure 32.9 Procedure for entering Global Operation mode



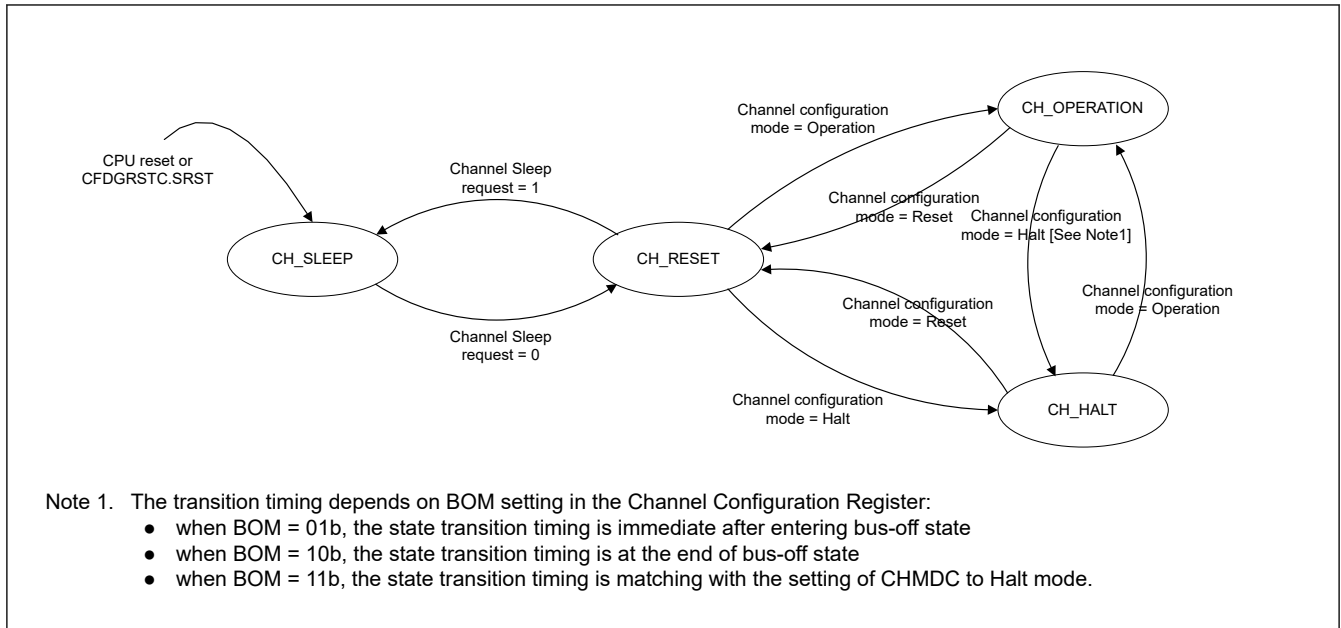
**Figure 32.10 Procedure for exiting Global Operation mode**

### 32.3.3 Channel Modes

A CAN channel can be in one of the following four channel modes:

- Reset
- Halt
- Operation
- Sleep.

Figure 32.11 shows the possible transitions between the channel modes.



**Figure 32.11 Transition between CAN channel modes**

### 32.3.3.1 CAN Channel Sleep Mode

After the release of a hardware reset or after setting and clearing the CFDGRSTC.SRST bit, a CAN channel of the CANFD module automatically enters Channel Sleep mode.

A CAN channel also enters Channel Sleep mode when the related Channel Sleep Mode Request bit is set while the CAN channel is in Channel Reset mode. Do not set this control bit in Channel Halt mode or Channel Operation mode.

Entering the CAN Channel Sleep mode instantly stops the clock supplied to the CAN channel unit and therefore reduces power consumption.

After setting the Channel Sleep Mode Request bit, it is necessary to confirm that the Channel Sleep mode status has been updated to indicate a successful transition to Channel Sleep mode before the Channel Sleep Mode Request bit can be cleared again.

During Channel Sleep mode, do not write to channel related registers. Read operation is still possible.

### 32.3.3.2 CAN Channel Reset Mode

A CANFD CAN channel enters this mode in the following ways:

- Channel Mode Control bit CFDC0CTR.CHMDC in the Channel Control Registers is configured for Channel Reset mode while the related CAN channel is in Channel Halt mode or Channel Operation mode
- Channel Sleep Mode Request bit is cleared while the related CAN channel is in Channel Sleep mode
- Global Mode Control bit CFDGCTR.GMDC is set to Global Reset mode and CAN channel is not in Channel Sleep mode or Channel Reset mode.

In Channel Reset mode, all CAN channel status and flag registers are initialized.

Additionally all channel related transmission control bits are cleared and the channel related TX Queue is disabled.

Configuration registers (except the Channel Test Mode registers) are not initialized in this mode and the CAN channel can be configured for communication.

See [section 32.3.4. Global Mode and Channel Mode Transition Interactions](#) for a detailed description of the behavior of all registers when transition to Channel Reset mode is performed.

After setting the Channel Mode Control bit CFDC0CTR.CHMDC to Channel Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDC0STS.CRSTSTS in the related Channel Status Registers has been updated to indicate a successful transition to Channel Reset mode before the related CFDC0CTR.CHMDC bit can be modified again.

See [Table 32.14](#) for the behavior of transitioning to Channel Reset mode while CAN communication is ongoing.

### 32.3.3.3 CAN Channel Halt Mode

A CANFD CAN channel enters this mode in the following ways:

- Channel Mode Control bit CFDC0CTR.CHMDC in the Channel Control Registers is configured for Channel Halt mode while the related CAN channel is in Channel Reset mode or Channel Operation mode
- Global Mode Control bit CFDGCTR.GMDC is set to Global Halt mode and CAN channel is in Channel Operation mode.

In Channel Halt mode, all channel CAN communication is suspended but all status and flag registers remain unchanged during Channel Halt mode entry (except for the bus-off case where REC and TEC values are cleared for this channel).

In addition, the Channel Test Mode Configuration and Control registers are not initialized in this mode.

The Channel Halt mode should be used to configure channel test modes.

See [section 32.3.4. Global Mode and Channel Mode Transition Interactions](#) for a detailed description of the behavior of all registers when transition to Channel Halt mode is performed.

After setting the Channel Mode Control bit CFDC0CTR.CHMDC to Channel Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDC0STS.CHLTSTS in the related Channel Status Register has been updated to indicate a successful transition to Channel Halt mode before the related CFDC0CTR.CHMDC can be modified again.

See [Table 32.14](#) for the transition behavior to Channel Halt mode while CAN communication is ongoing.

**Table 32.14 Transition behavior in CAN Reset mode and Halt mode**

Mode	State		
	Receiver	Transmitter	Bus-Off
<b>CAN Channel Reset mode (CFDC0CTR.CHMDC = 01b)</b>	The CAN channel enters Channel Reset mode without waiting for the completion of the ongoing reception.*1	The CAN channel enters Channel Reset mode without waiting for the completion of the ongoing transmission.*1	The CAN channel enters Channel Reset mode without waiting for the completion of the bus-off recovery.
<b>CAN Channel Halt mode (CFDC0CTR.CHMDC = 10b)</b>	CAN channel enters Channel Halt mode at the end of the ongoing reception or error.*2	CAN channel enters Channel Halt mode after completion of the ongoing transmission.	When CFDC0CTR.BOM is set to 00b, a Channel Halt mode request is accepted only after the completion of the full bus-off recovery sequence. When CFDC0CTR.BOM is set to 10b, the CAN channel transits automatically to Channel Halt mode after waiting for the completion of the bus-off recovery. When CFDC0CTR.BOM is set to 01b, the CAN channel transits automatically to Channel Halt mode without waiting for the completion of the bus-off recovery. When CFDC0CTR.BOM is set to 11b, the CAN channel enters Channel Halt mode as soon as Channel Halt mode is requested (without waiting for the completion of the bus-off recovery).

Note 1. If the entry to Channel Reset mode is required only at the end of an ongoing communication, then Channel Halt mode can be requested first to prevent interruption of CAN communication by direct transition to Channel Reset mode. After the CAN channel enters Channel Halt mode, the Channel Reset mode can be requested.

Note 2. If CAN communication is locked at dominant level after an error flag, software can detect this situation by monitoring the channel related BusLock flag and resolve lock condition by setting the CAN channel to Channel Reset mode.

### 32.3.3.4 CAN Channel Operation Mode

The Channel Operation mode is activated by setting the CFDC0CTR.CHMDC bits to 00b. If 11 consecutive recessive bits are detected after entering the CAN Operation mode, the CFDC0STS.COMSTS bit is set and the CAN channel:

- Enables the functions of the channel communication by allowing the channel to become an active node on the CAN network
- Releases the internal fault confinement logic including receive and transmit error counters

At this point, the CAN channel can start transmission and reception of CAN messages.

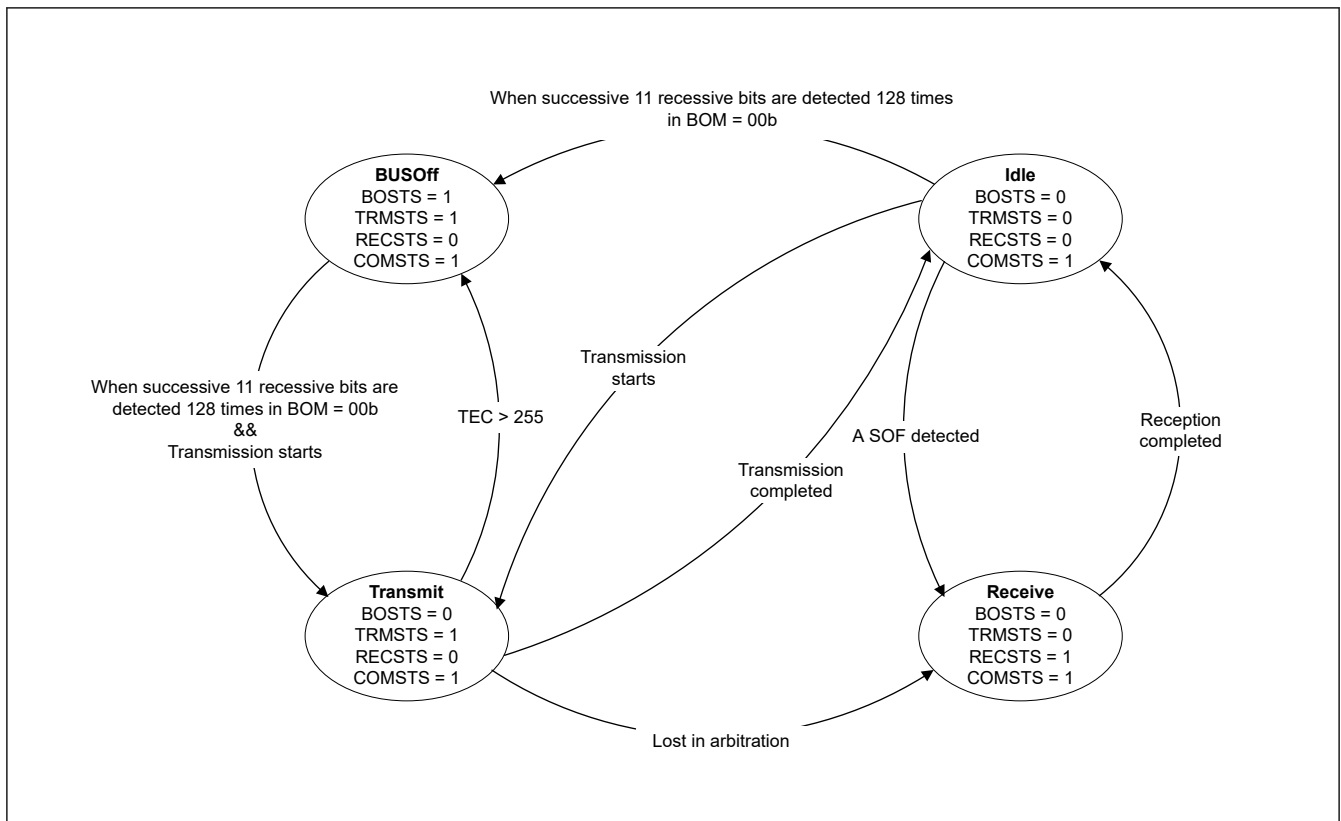
Within the CAN Channel Operation mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (see [Figure 32.12](#)):

- Channel idle: The CAN channel is neither receiving nor transmitting
- Channel receives: The channel is receiving a CAN message sent by another CAN node
- Channel transmits: The channel is transmitting a CAN message

Note: The channel may receive its own message simultaneously when Self-test mode is enabled.

- Channel is in bus-off state: The CAN channel is cut-off from CAN bus communication.

After setting the Channel Mode Control bit CFDC0CTR.CHMDC to Channel Operation mode, it is necessary to confirm that the Channel Reset Mode Status bit CFDC0STS.CRSTSTS and the Channel Halt Mode Status bit CFDC0STS.CHLTSTS in the Channel Status Register have been updated to indicate a successful transition to Channel Operation mode before the related CFDC0CTR.CHMDC bit can be changed again.



**Figure 32.12 Sub-modes of CAN Channel Operation mode (only when BOM = 00b)**

### 32.3.3.5 CAN Channel Bus-Off State

The CAN channel bus-off state is entered according to the fault confinement rules of the CAN specification. The following modes can be configured for returning to the CAN Channel Operation mode from the bus-off state:

- CFDC0CTR.BOM = 00b:  
Bus-Off recovery is compliant to ISO 11898-1, namely the CAN channel re-enters CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. TEC and REC counters are initialized to 0. The Bus-Off Recovery Flag CFDC0ERFL.BORF is set in this case.
- CFDC0CTR.BOM = 01b:

The CAN channel changes the value of the CFDC0CTR.CHMDC bits within the CAN Channel Control Register to 10b and switches immediately to Channel Halt mode automatically after entering bus-off state. TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDC0ERFL.BORF is not set in this case.

- CFDC0CTR.BOM = 10b:  
The CAN channel changes the value of the CFDC0CTR.CHMDC bits within the CAN Channel Control Register to 10b as soon as it reaches bus-off state and enters Channel Halt mode automatically after the CAN channel has completed the bus-off recovery sequence (after 11 consecutive recessive bits are detected 128 times). TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDC0ERFL.BORF is set in this case.
- CFDC0CTR.BOM = 11b:  
Bus-off recovery is initiated but CAN channel can immediately enter Channel Halt mode when still in bus-off state if a request is made to enter Channel Halt mode.  
TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDC0ERFL.BORF is not set.  
Without setting CFDC0CTR.CHMDC [1:0] = 10b and when 11 recessive bits is detected 128 times continuously, transition conditions become the same as CFDC0CTR.BOM = 00b.

Note: If the recovery from bus-off occurs normally in this mode (after waiting for 128 sequences of 11 consecutive recessive bits), and no halt request has been generated during this period, then the Bus-Off Recovery flag CFDC0ERFL.BORF is set.

When software writes to the CFDC0CTR.CHMDC bit at the same time as the CAN channel enters Halt mode (at the start of bus-off when CFDC0CTR.BOM = 01b, or at the end of bus-off when CFDC0CTR.BOM = 10b), the software request has the highest priority.

Note: In the above case, the automatic setting of the CFDC0CTR.CHMDC bit to Channel Halt mode request is performed when the CFDC0CTR.CHMDC bit value is previously 00b (Channel Operation mode).

Additionally, it is possible to force the CAN channel to recover from the bus-off state by setting CFDC0CTR.RTBO to 1. The error state changes from bus-off state to integrating state with a maximum delay of 1 CAN bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The Bus-Off Recovery Flag is not set in this case, and the TEC and REC counters are initialized to 0.

Before setting CFDC0CTR.RTBO to 1, all pending transmissions from the TX message buffers, TX Queues and/or Common FIFO in TX mode should be disabled.

The disable of the pending transmission message buffer, TX Queue or FIFO must be confirmed by the corresponding acknowledge flags.

For the TX message buffer, the acknowledge flags are the Transmission Result Flags (CFDTMSTSj.TMTRF). For the TX Queue, it is the TX Queue Empty flag (CFDTXQSTS.TXQEMP). For the FIFO, it is the FIFO Empty flag (CFDCFSTS.CFEMP).

The CFDC0CTR.RTBO bit should be used for bus-off recovery only when CFDC0CTR.BOM is set to 00b.

Setting this bit in any state other than bus-off has no effect and the bit is cleared immediately.

Table 32.15 shows the settings for the Bus-Off Entry flag CFDC0ERFL.BOEF and the Bus-Off Recovery flag CFDC0ERFL.BORF for the different configurations of CFDC0CTR.BOM.

**Table 32.15 Behavior of Bus-off Entry and Recovery flags**

BOM	BOEF bit set	BORF bit set
00b	Always (on entry to bus-off)	Always (on exit from bus-off)
00b CFDC0CTR.RTBO set to 1	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software sets CFDC0CTR.RTBO to 1'
01b	Always (on entry to bus-off)	Never
10b	Always (on entry to bus-off)	Always (on exit from bus-off)
11b	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software issues a Halt request

For an efficient software procedure, it is not necessary to wait for the bus-off recovery sequence to end.

It is possible to perform a transmission re-initialization during bus-off recovery. To do this, follow the recommended software flow in [Figure 32.13](#).

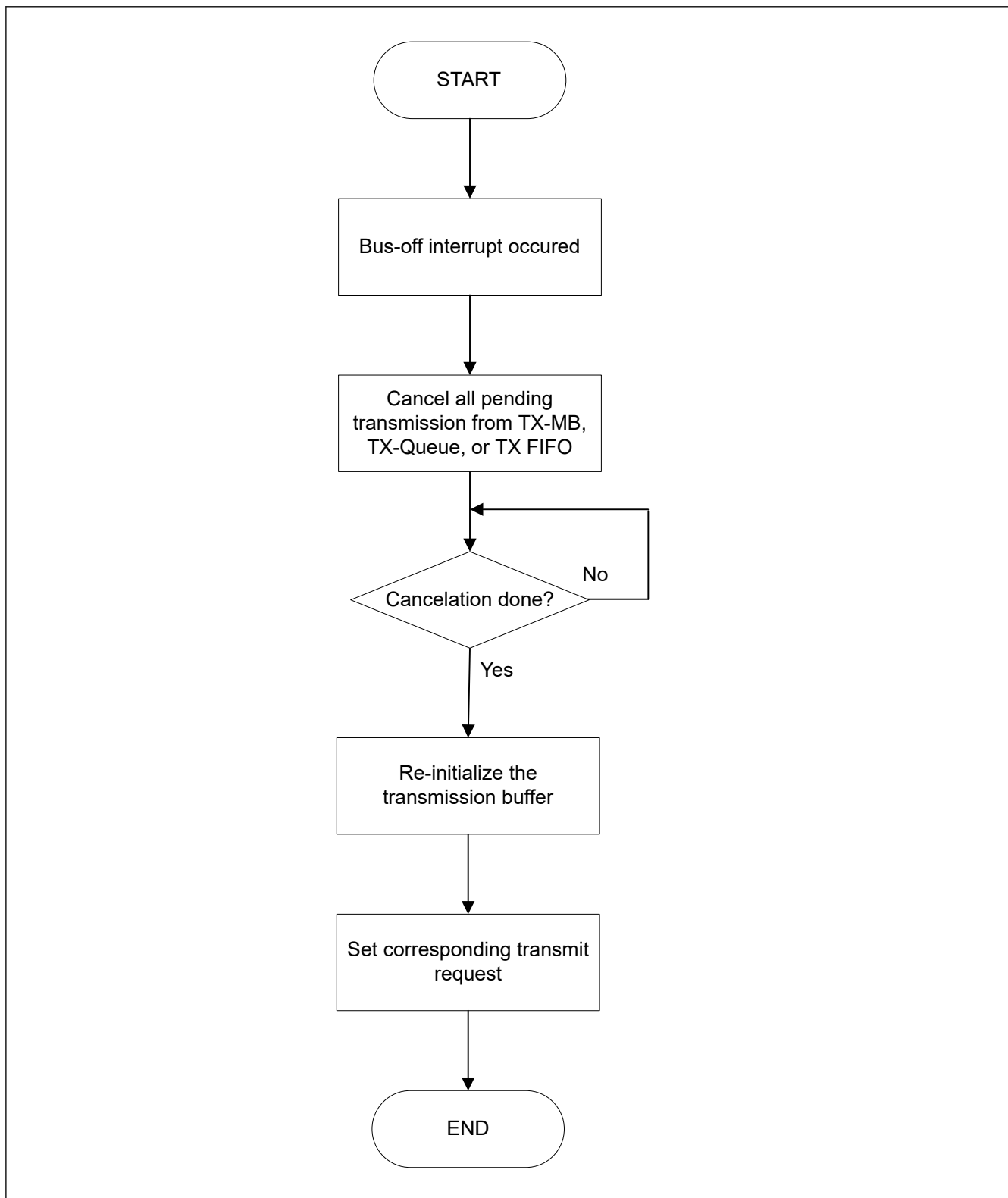


Figure 32.13 Transmission re-initialization during bus-off

### 32.3.4 Global Mode and Channel Mode Transition Interactions

The interaction between Global mode setting and Channel mode setting is as follows:



- Changing the Channel Mode Control bit CFDC0CTR.CHMDC in the Channel Control Registers does not affect the Global Mode Control bit CFDGCTR.GMDC.
- Changing the Global Mode Control bit CFDGCTR.GMDC affects the channel mode control as described in [Table 32.16](#).

**Table 32.16 Interaction between Global and Channel mode transition**

Global mode change	Channel mode	Channel mode transition action
Sleep → Reset	Sleep	Channel remains in Sleep mode
Sleep → Halt	— (Global mode change not possible)	
Sleep → Operation	— (Global mode change not possible)	
Reset → Sleep	Sleep	Channel remains in Sleep mode
	Reset	Channel Sleep request bit is set automatically, channel enters Sleep Mode
Reset → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Reset → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Halt → Sleep	— (Global mode change not possible)	
Halt → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel enters Reset mode
Halt → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
Operation → Sleep	— (Global mode change not possible)	
Operation → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel enters Reset mode
	Operation	Channel mode control is set to Reset mode, channel enters Reset mode
Operation → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
	Operation	Channel mode control is set to Halt mode, channel enters Halt mode after communication finished

### 32.3.4.1 Timing of Global Mode Change

The transition time for the Global mode changes are shown in the following table.

**Table 32.17 Maximum transition time for the global mode (1 of 2)**

From	To	Maximum transition time
GL_SLEEP	GL_RESET	3 peripheral clock cycles*2
GL_RESET	GL_SLEEP	3 peripheral clock cycles
GL_RESET	GL_HALT	10 peripheral clock cycles
GL_RESET	GL_OPERATION	10 peripheral clock cycles

**Table 32.17** Maximum transition time for the global mode (2 of 2)

From	To	Maximum transition time
GL_HALT	GL_RESET	2 CAN bit times
GL_HALT	GL_OPERATION	3 peripheral clock cycles
GL_OPERATION	GL_RESET	2 CAN bit times
GL_OPERATION	GL_HALT	3 CAN frames <sup>*1 *3</sup>

Note 1. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.

Note 2. Exit GL\_SLEEP mode only when CFDGSTS.GRAMINIT is cleared.

Note 3. TQ, CAN frame and CAN bits are related to the individual channels. For the maximum transition time, the channel with the lowest baud rate must be used.

### 32.3.4.2 Timing of Channel Mode Change

The transition time for the Channel mode changes are shown in the following table.

**Table 32.18** Maximum transition time for the channel mode

From	To	max. transition time
CH_SLEEP	CH_RESET	3 peripheral clock cycles
CH_RESET	CH_SLEEP	3 peripheral clock cycles
CH_RESET	CH_HALT	3 CAN bit times
CH_RESET	CH_OPERATION	4 CAN bit times
CH_HALT	CH_RESET	2 CAN bit times
CH_HALT	CH_OPERATION	4 CAN bit times <sup>*3</sup>
CH_OPERATION	CH_RESET	2 CAN bit times
CH_OPERATION	CH_HALT	2 CAN frames <sup>*1 *2</sup>

Note 1. The time specified for this transition does not include the case where channel enters bus-off state. For bus-off, the timing depends on the configuration of the CFDC0CTR.BOM[1:0] bits.

Note 2. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.

Note 3. In general, if the baudrate prescaler value CFDC0NCFG.NBRP is changed in CH\_HALT mode, the transition time can deviate. The internal prescaler is a free running down counter that creates the TQ clock, and new BRP value is captured when the counter reaches the value 0.

## 32.4 Initialization

Before joining CAN communications, configure the following settings:

- Clock setting
- Bit timing setting (nominal and data rate)
- Baud Rate setting (nominal and data rate)
- CANFD setting
- Acceptance Filter setting (configuration of Global Acceptance Filter List)
- Reception, Transmission and GW-FIFO setting
- CAN Operation mode setting

### 32.4.1 Initialization of CAN Clock, Bit Timing and Baud Rate

#### 32.4.1.1 Bit Timing Conditions

The following lines describe the composition of each segment and the restriction that apply to the segment setting.

1. Each segment setting  
SS = Fixed to 1 TQ

TSEG1 = See to (CFDC0NCFG) and (CFDC0DCFG)<sup>\*1</sup>

TSEG2 = See to (CFDC0NCFG) and (CFDC0DCFG)<sup>\*1</sup>

SJW = See to (CFDC0NCFG) and (CFDC0DCFG)<sup>\*1</sup>

SS + TSEG1 + TSEG2 = 5 to 49 TQs for Data Bit Rate and 8 to 385 for Nominal Bit Rate

2. Restriction on TSEG1, TSEG2 and SJW

TSEG1(N) > TSEG2(N) ≥ SJW(N)

TSEG1(D) ≥ TSEG2(D) ≥ SJW(D)<sup>\*1</sup>

When only classical frames are used, configure the bit fields TSEG1 and TSEG2 of CFDC0DCFG to valid values.

Note 1. This feature is not available in the classical CAN function.

Table 32.19 shows an example of how to set the bit timing to achieve the required Sample Point settings.

Table 32.19 Bit timing examples

1 bit	Set value (TQ)				Sample point <sup>*1</sup> (%)
	SS	TSEG1	TSEG2	SJW	
5TQ	1	2	2	1	60.00
8TQ	1	4	3	1	62.50
	1	5	2	1	75.00
10TQ	1	6	3	1	70.00
	1	7	2	1	80.00
12TQ	1	8	3	1	75.00
	1	9	2	1	83.33
15TQ	1	10	4	1	73.33
	1	11	3	1	80.00
16TQ	1	10	5	1	68.75
	1	11	4	1	75.00
20TQ	1	12	7	1	65.00
	1	13	6	1	70.00
24TQ	1	15	8	1	66.66
	1	16	7	1	70.83
50TQ	1	39	10	4	80.00

Note 1. Sample point (in case of 75%)

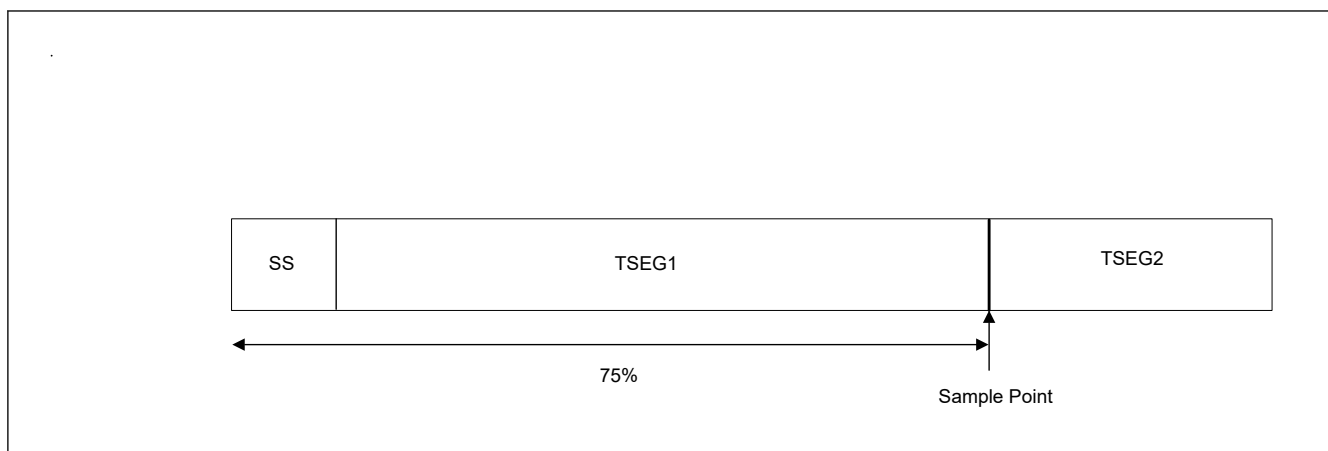


Figure 32.14 Sample point (in case of 75%)

### 32.4.1.2 CAN Bit Timing

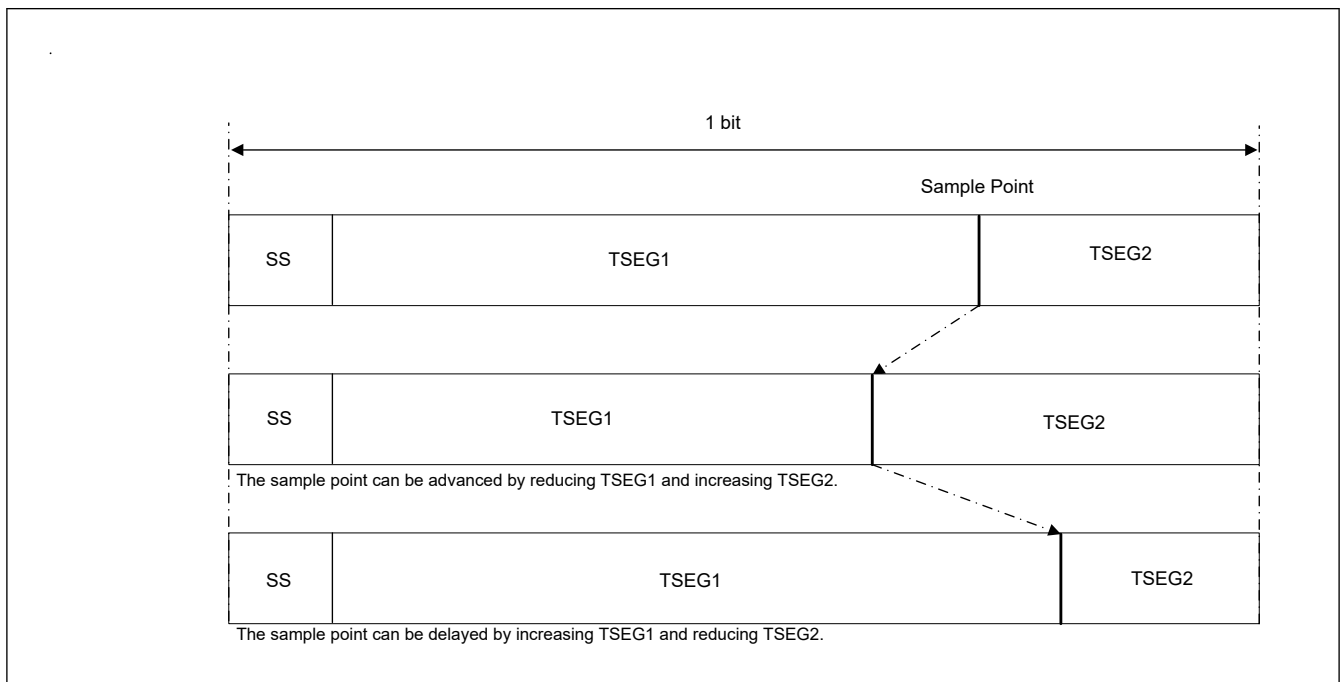
In the CAN protocol, each bit in a communication frame is composed of three segments that can be configured individually for channel using the related CFDC0NCFG and CFDC0DCFG\*1 registers.

Note 1. This register is not available in the classical CAN function.

Figure 32.15 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (TQ), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the baud rate prescaler (nominal and data rate).



**Figure 32.15 Segment composition of a bit and the sample point**

1. SS: Synchronization Segment  
This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the interframe space. This comprises of intermission, suspend transmission, bus idle, during bus idle, and all nodes that can start transmission.
2. TSEG1: Time Segment 1  
This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.
3. TSEG2: Time Segment 2  
This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW. While sending or receiving a message, communication frames between some nodes may get out of sync due to a drift in the oscillator frequency or a delay in the transmission path. This is referred to as a phase error.
4. SJW: Resynchronization Jump Width  
This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

Figure 32.15 shows only one symbolic sample point.

### 32.4.1.3 Baud Rate

Either the CANFD core clock (CANFDCLK) or the external oscillator clock (CANMCLK) can be selected globally as CAN communication clock.

The transfer speed is determined by the DLL clock, the divide-by-N value of the baud rate prescaler, and the number of TQs in one bit.

$$\text{baudrate} = \frac{\text{DLL Clock}}{(\text{number\_of\_time\_quanta\_per\_bit}) \times (\text{BRP} + 1)}$$

Figure 32.16 shows a block diagram of the circuit that generates the CAN channel communication clock and Table 32.20 shows a baud rate examples.

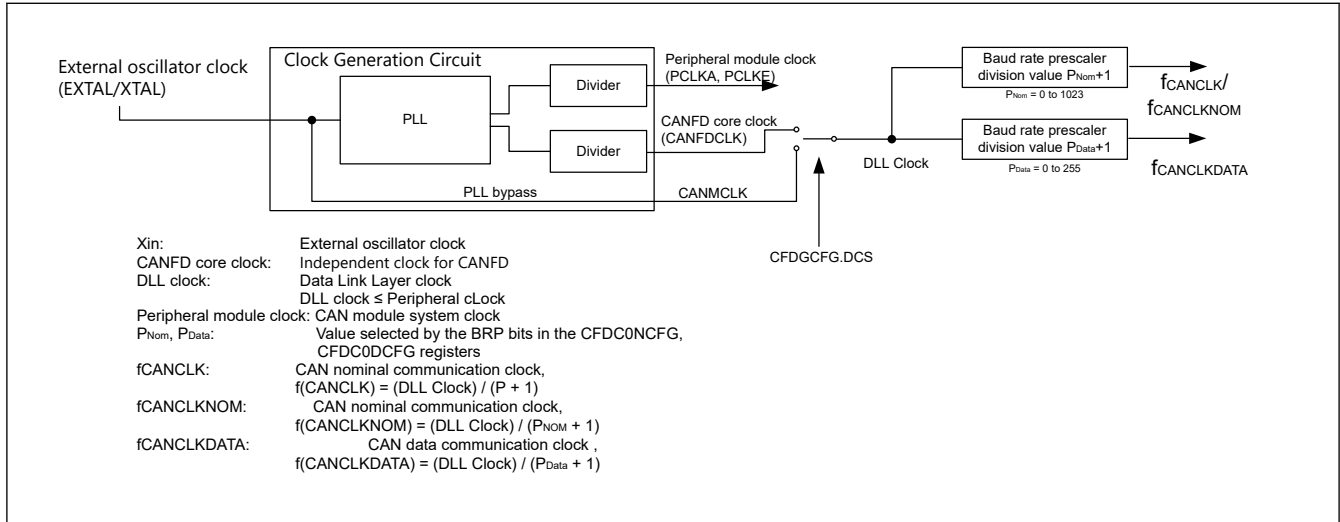


Figure 32.16 Block diagram of the circuit that generates the CAN channel communication clock

Table 32.20 Nominal baud rate calculation formula and example CAN communication configurations

Baud rate calculation formula	(DLL clock) (baud rate prescaler divide-by-N value*1) × (number of TQs in one bit)								
	80 MHz	40 MHz	32 MHz	30 MHz	24 MHz	20 MHz	16 MHz	10 MHz	8 MHz*2
1 Mbps	8TQ (10) 20TQ (4)	8TQ (5) 20TQ (2)	8TQ (4) 16TQ (2)	10TQ (3) 15TQ (2)	8TQ (3) 12TQ (2) 24TQ (1)	10TQ (2) 20TQ (1)	8TQ (2) 16TQ (1)	10TQ (1)	8TQ (1)
500 Kbps	8TQ (20) 20TQ (8)	8TQ (10) 20TQ (4)	8TQ (8) 16TQ (4)	10TQ (6) 15TQ (4) 20TQ (3)	8TQ (6) 12TQ (4) 24TQ (2)	10TQ (4) 20TQ (2)	8TQ (4) 16TQ (2)	10TQ (2) 20TQ (1)	8TQ (2) 16TQ (1)
250 Kbps	8TQ (40) 20TQ (16)	8TQ (20) 20TQ (8)	8TQ (16) 16TQ (8)	10TQ (12) 15TQ (8) 20TQ (6)	8TQ (12) 12TQ (8) 24TQ (4)	10TQ (8) 20TQ (4)	8TQ (8) 16TQ (4)	10TQ (4) 20TQ (2)	8TQ (4) 16TQ (2)
125 Kbps	8TQ (80) 20TQ (32)	8TQ (40) 20TQ (16)	8TQ (32) 16TQ (16)	10TQ (24) 15TQ (16) 20TQ (12)	8TQ (24) 12TQ (16) 24TQ (8)	10TQ (16) 20TQ (8)	8TQ (16) 16TQ (8)	10TQ (8) 20TQ (4)	8TQ (8) 16TQ (4)
83.3 Kbps	8TQ (120) 12TQ (80) 16TQ (60) 24TQ (40)	8TQ (60) 12TQ (40) 16TQ (30) 24TQ (20)	8TQ (48) 12TQ (32) 16TQ (24) 24TQ (16)	8TQ (45) 10TQ (36) 12TQ (30) 15TQ (24) 20TQ (18) 24TQ (15)	8TQ (36) 12TQ (24) 16TQ (18) 24TQ (12)	8TQ (30) 10TQ (24) 12TQ (20) 15TQ (16) 16TQ (15) 20TQ (12) 24TQ (10)	8TQ (24) 12TQ (16) 16TQ (12) 24TQ (8)	8TQ (15) 10TQ (12) 12TQ (10) 15TQ (8) 20TQ (6) 24TQ (5)	8TQ (12)
33.3 Kbps	8TQ (300) 12TQ (200) 16TQ (150) 20TQ (120) 24TQ (100)	8TQ (150) 12TQ (100) 16TQ (75) 20TQ (60) 24TQ (50)	8TQ (120) 10TQ (96) 12TQ (80) 15TQ (64) 16TQ (60) 20TQ (48) 24TQ (40)	10TQ (90) 12TQ (75) 15TQ (60) 20TQ (45)	8TQ (90) 10TQ (72) 12TQ (60) 15TQ (48) 16TQ (45) 20TQ (36) 24TQ (30)	8TQ (75) 10TQ (60) 12TQ (50) 15TQ (40) 16TQ (30) 20TQ (25)	8TQ (60) 10TQ (48) 12TQ (40) 15TQ (32) 16TQ (30) 20TQ (24) 24TQ (20)	10TQ (30) 12TQ (25) 15TQ (20) 20TQ (15)	8TQ (30)

Note: Shown in ( ) are the baud rate prescaler divide-by-N value.  
 Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 - 1023) P: value selected by the BRP bits in the Channel Configuration Registers.  
 Note 2. Minimum frequency to achieve maximum nominal baud rate of 1 Mbps.

**Table 32.21 Baud rate calculation example for nominal and data bit rate CAN communication configurations**

Baud rate calculation formula	(DLL clock) (baud rate prescaler divide-by-N value*1) × (number of TQs in one bit)		
	80 MHz	40 MHz	20 MHz
Nominal 1 Mbps Data 8 Mbps	80TQ (1)	40TQ (1)	20TQ (1)
	10TQ (1)	5TQ (1)	Not possible
Nominal 1 Mbps Data 5 Mbps	80TQ (1)	40TQ (1)	20TQ (1)
	16TQ (1)	8TQ (1)	Not possible
Nominal 500 Kbps Data 2 Mbps	160TQ (1)	80TQ (1)	40TQ (1)
	40TQ (1)	20TQ (1)	10TQ (1)

Note: Shown in ( ) are the baud rate prescaler divide-by-N values and this table is not available in the classical CAN function.

Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 - 1023) P: value selected by the BRP bits in the Channel Configuration Registers.

For optimum clock tolerance in networks using the FD frame format, the length of the time quantum should be the same in nominal bit time and in data bit time. This means CFDC0NCFG.NBRP = CFDC0DCFG.DBRP.

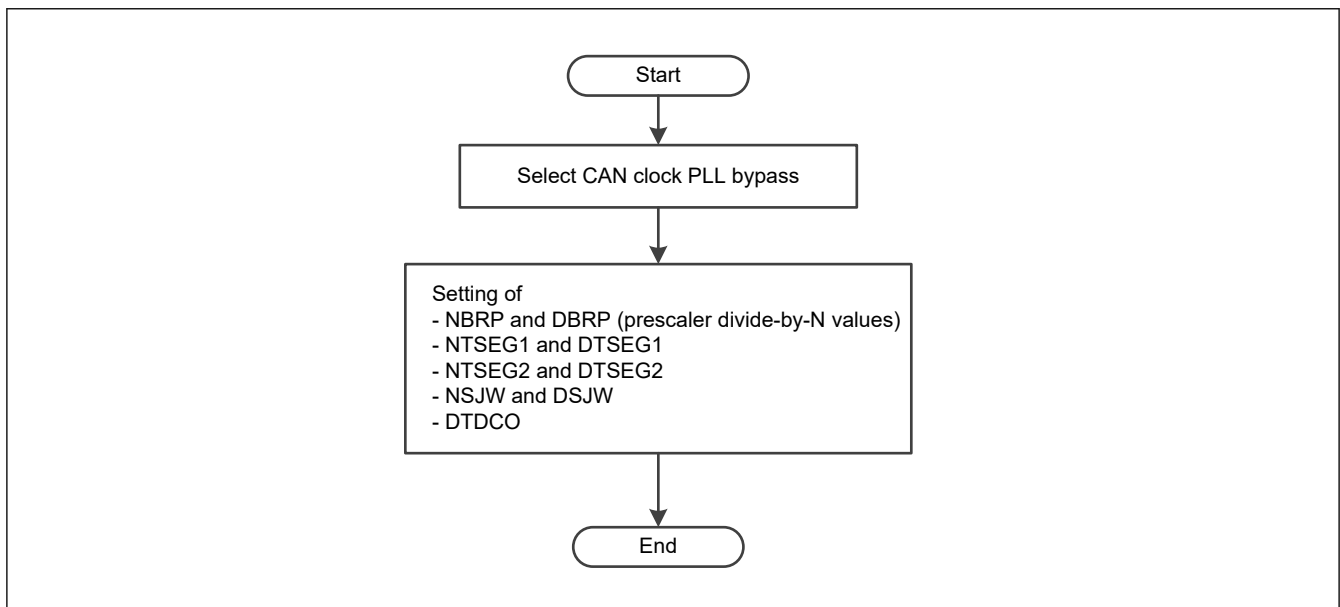
Additionally, if transceiver delay compensation is used, do not program the CFDC0DCFG.DBRP bit to be greater than 1, as 1 means divide by 2.

### 32.4.1.4 Setting of CAN Clock, Bit Timing and Baud Rate

Figure 32.17 shows the procedure for setting the CAN clock and the baud rate for each channel.

These settings should be performed during Channel Reset mode (Configuration mode) for the CAN channels.

Before going to channel communication state, the baud rate must be configured, otherwise the mode does not switch correctly.



**Figure 32.17 Procedure for setting the CAN bit timing and baud rate**

### 32.4.1.5 Transmitter Delay Compensation

This chapter is not valid for classical CAN.

When a high baud rate is used such as 5 to 8 Mbps for the data phase, the transmitter delay can become greater than TSEG1. In this case, the transmitter always detects a bit-error in the data phase of the CANFD frame. The TDC compensates for the inability of the transmitter to receive its own transmitted bit at the sample point of that bit.

There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the data phase of CANFD frames. This is derived from the Transceiver Delay Compensation Result bit (CFDC0FDSTS.TDCR) as shown in Figure 32.18.

The resolution of the configuration, measured and offset values is based on the CAN channel DLL clock.

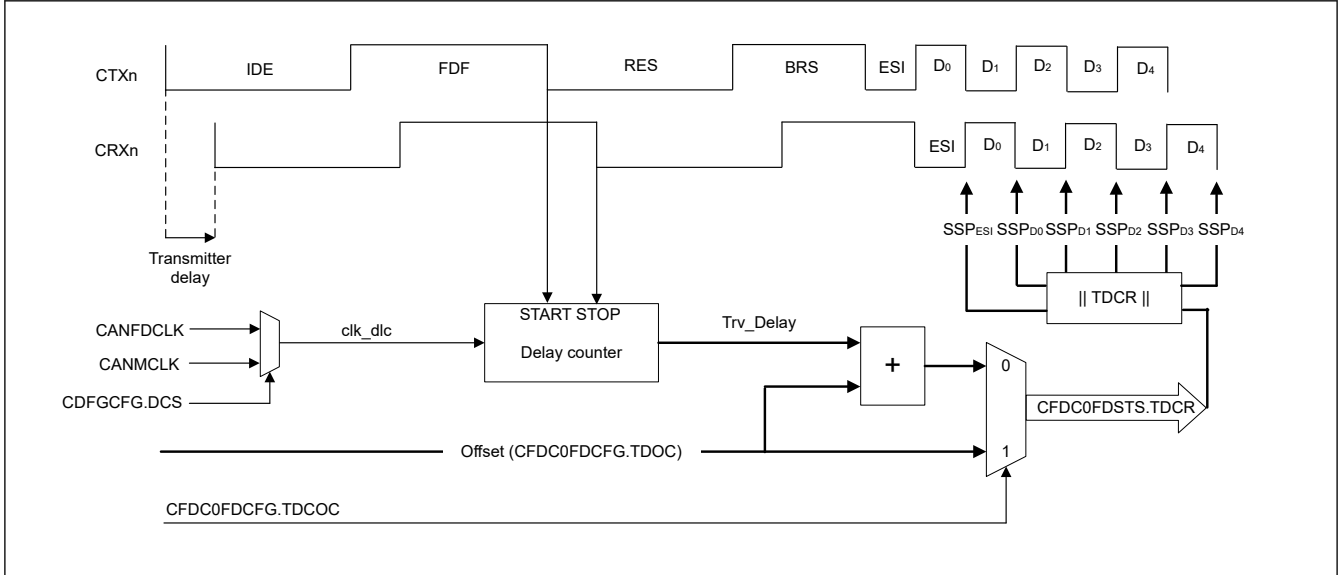


Figure 32.18 Transmitter delay compensation

The measured Trv\_Delay is based on the number of clk\_dlc clock cycles. The delay is counted up by one for each started clock until the dominant value is seen on CRXn. Figure 32.19 shows the measured result. Trv\_Delay counted to maximum 127 with a clk\_dlc clock.

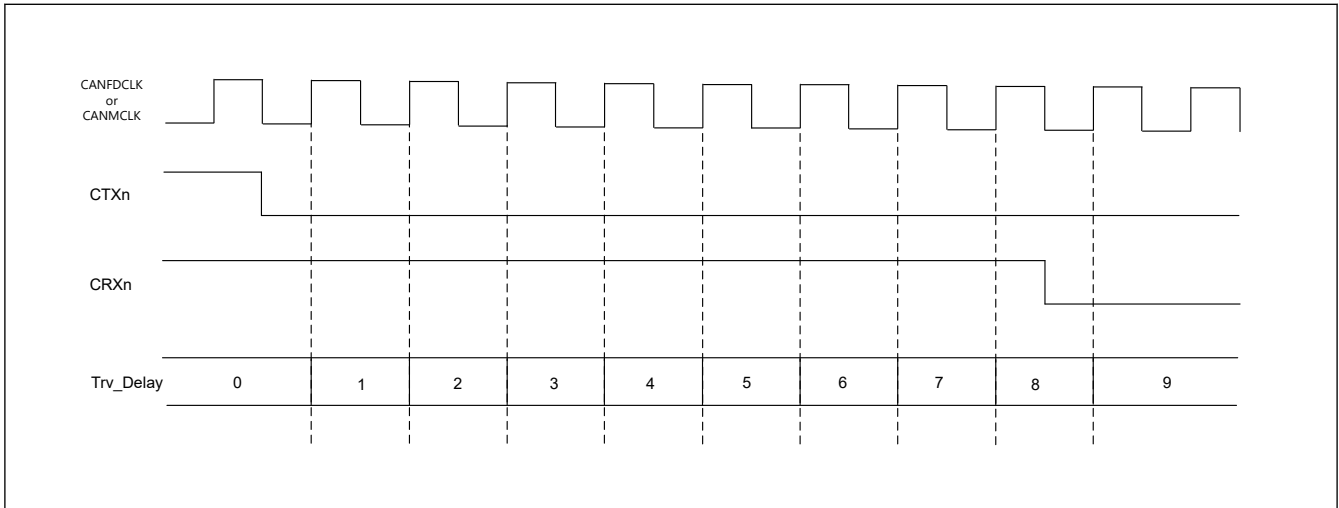
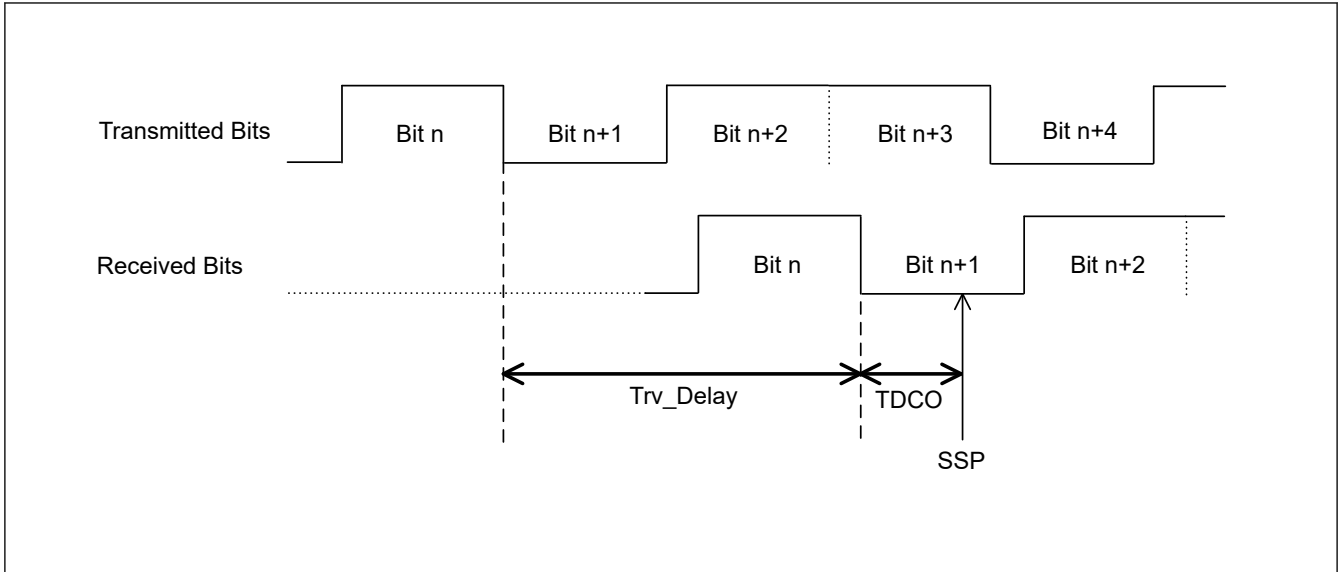


Figure 32.19 Trv\_Delay measurement example

The SSP is calculated by taking the result from CFDC0FDSTS.TDCR and rounding the value down to the nearest integer number of data time quanta.

Figure 32.20 shows the positioning of the secondary sample point. When CFDC0FDCFG.TDCOC is equal to 0, the SSP is equal to the Trv\_Delay (measured delay) + CFDC0FDCFG.TDCO, rounded down to the nearest integer number of time quanta. Usually, the TDCO value should have the size of (SyncSegmentdata + TSEG1data) to position the SSP to a theoretical location of the sample point.

If the CFDC0FDCFG.TDCOC is equal to 1, the SSP is defined by CFDC0FDCFG.TDCO. If CFDC0DCFG.DBRP is greater than 0, the value is also rounded down to the nearest integer number of time quanta.



**Figure 32.20 Position of the secondary sample point**

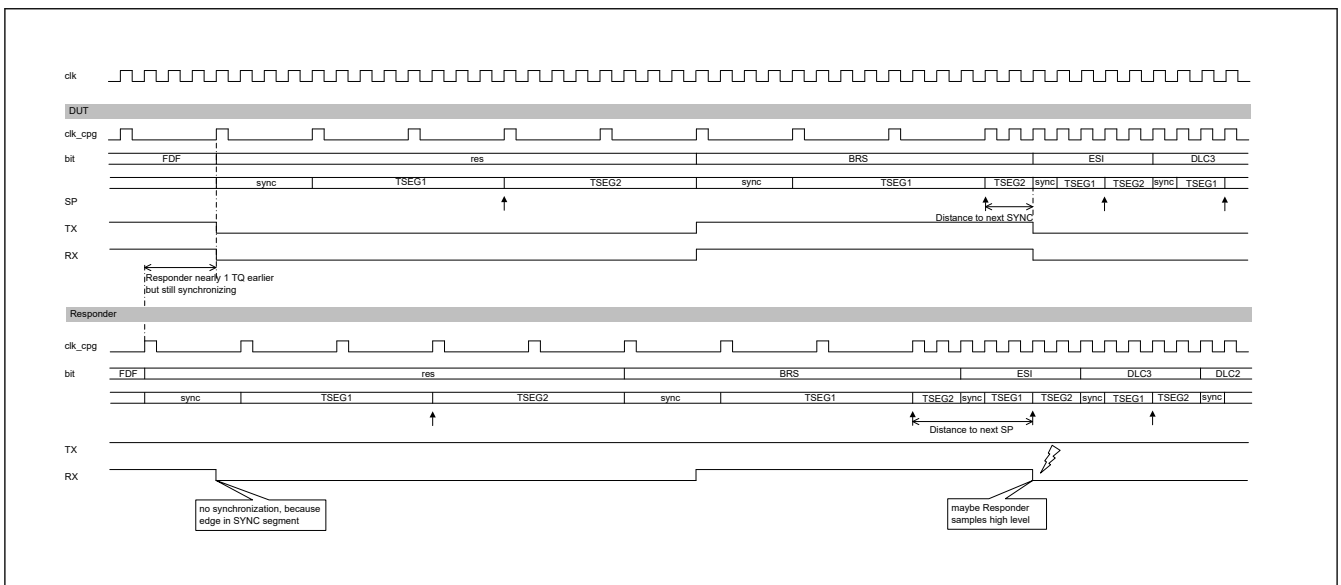
The maximum delay ( $Trv\_Delay + TDCO$ ) which can be compensated by the CANFD module is  $(6 \text{ data bits} - 2clk\_dlc)$ .

The ISO 11898-1 allows you to set different values for  $BRP\_data$  and  $BRP\_nom$ .

If different values are used for  $CFDC0NCFG.NBRP$  and  $CFDC0DCFG.DBRP$ , then two CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after sample point of the BRS bit. This condition is shown in [Figure 32.21](#).

The length of the time quantum should be the same in the nominal bit time and in the data bit time. This means  $CFDC0NCFG.NBRP = CFDC0DCFG.DBRP$ .

Different bit rates can be achieved by selecting different configuration values for the Time Segments. The nominal bit rate can be configured from 8 to 385 TQs and the data bit rate from 5 to 49 TQs.



**Figure 32.21 Loss of synchronization between two CAN nodes**

The transmitter delay compensation measurement result is updated at the falling edge from FDF bit to RES bit when configured accordingly ( $CFDC0FDCFG.TDCE = 1$ ,  $CFDC0FDCFG.TDCOC = 0$ ).

[Figure 32.22](#) shows the read flow to get the measured transmitter delay compensation result.



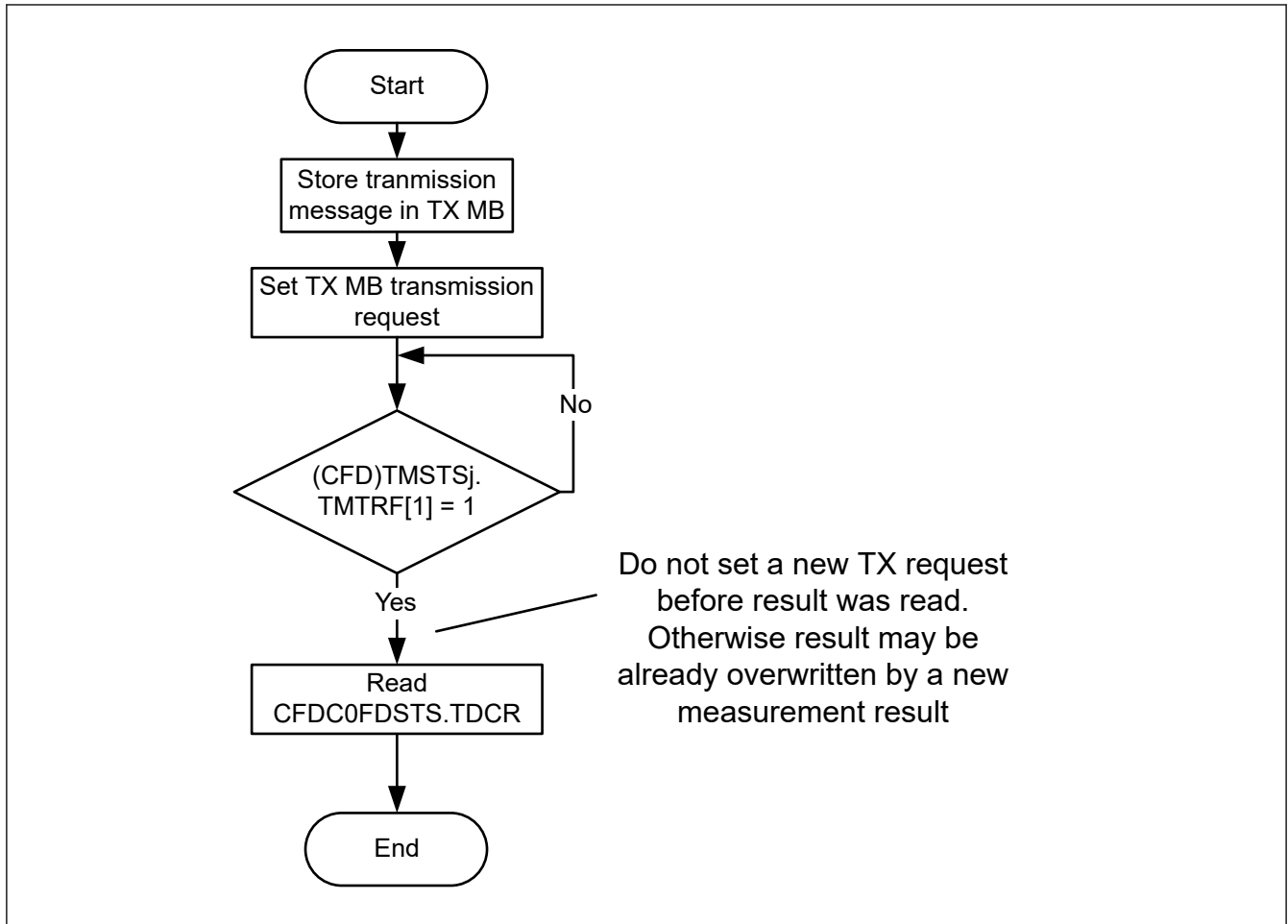


Figure 32.22 TDC result read flow

### 32.4.2 CAN Module Configuration after Hardware Reset

After a hardware reset (power on reset) or after setting and clearing a `CFDGRSTC.SRST` bit, the CANFD module enters Global Sleep mode automatically.

To enable configuration of the CANFD module, you must exit Sleep mode by clearing the Global Sleep Request bit `CFDGCTR.GSLPR` to 0.

After a hardware reset, the module starts RAM initialization, the `CFDGSTS.GRAMINIT` bit in the Global Status Register is set automatically to indicate that the CANFD logic is initializing the RAM.

After RAM initialization is complete, this bit is cleared automatically.

RAM initialization is necessary to avoid setting of false ECC error flag after HW reset the random data presented in the RAM.

Do not access registers of CANFD in either read or write until RAM initialization is complete and the `CFDGSTS.GRAMINIT` bit is cleared.

Before going to communication mode, the Global Acceptance Filter List and message FIFO buffers must be configured. In addition, CAN channel must be configured such as CAN bit timing. For this configuration, CAN channel must be released from Channel Sleep mode and must be configured for communication in Channel Reset mode (Configuration mode).

Figure 32.23 shows the configuration procedure. For details about each step, see [section 32.5. Acceptance Filtering Function using Global Acceptance Filter List \(AFL\)](#), [section 32.6. FIFO Buffers and Normal Message Buffer Configuration](#), [section 32.7. Interrupts and DMA](#) and [section 32.4.1.3. Baud Rate](#).

The CANFD module does not perform the RAM initialization sequence after executing a software reset by setting `CFDGRSTC.SRST`.

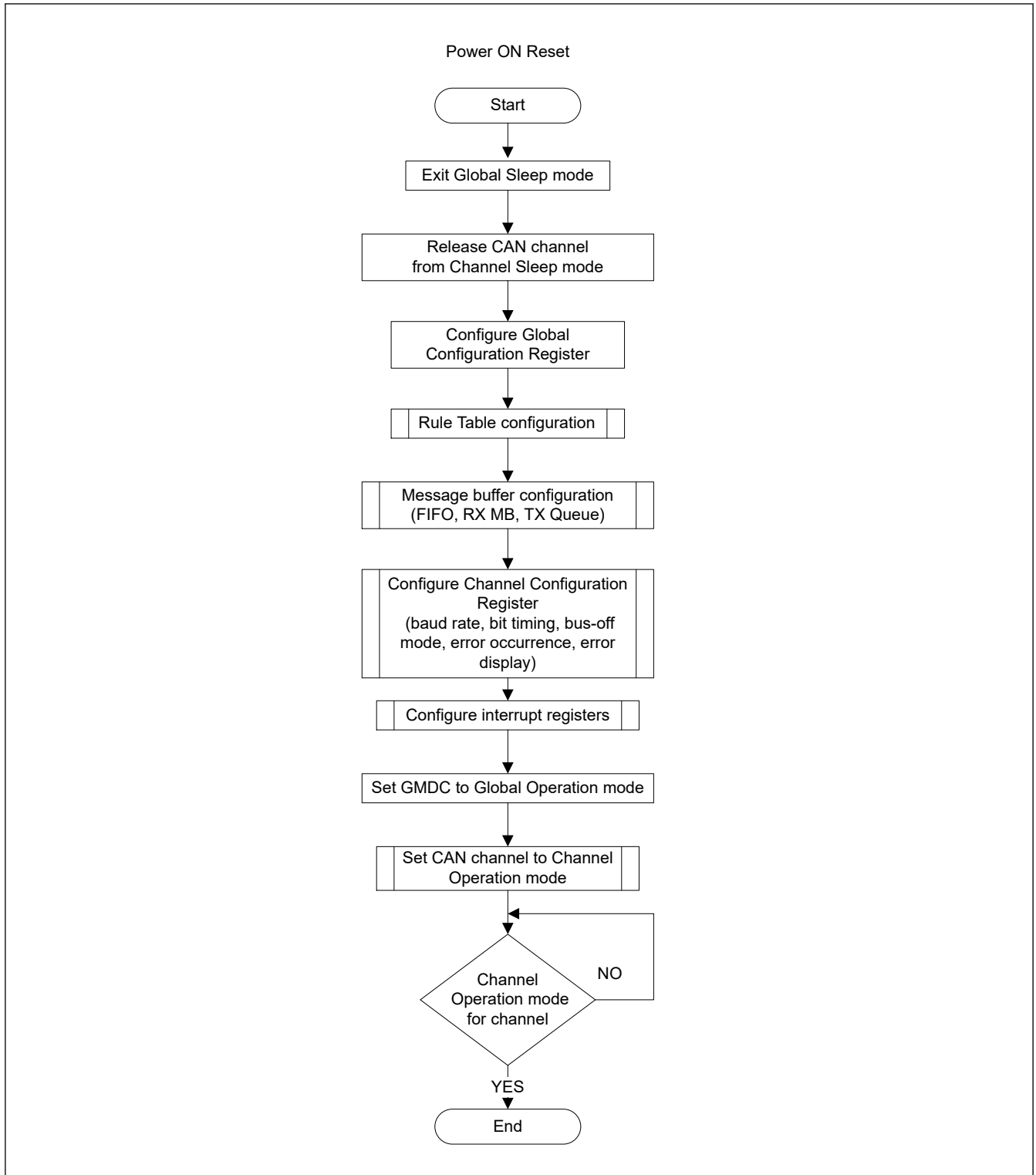


Figure 32.23 Configuration procedure after a hardware reset

## 32.5 Acceptance Filtering Function using Global Acceptance Filter List (AFL)

### 32.5.1 Overview

The CANFD module can handle message acceptance filtering with a global Acceptance Filter List (called AFL). Each element of the AFL defines a filter rule for messages received on a specific channel.

The following actions are performed based on the AFL entries:

- Acceptance filtering based on received CAN Identifier and masking
- DLC filtering based on received DLC value
- Message data payload according to the `CFDGCFCG.CMPOC` bit\*<sup>1</sup>
- Storage of accepted messages in the message buffer objects defined in the related AFL entry
- Attaching a 16-bit pointer to the stored messages defined in the related AFL entry, for example to support AUTOSAR applications
- Attaching a 2-bit information label to the stored messages defined in the related AFL entry

Note 1. This feature is not available in the classical CAN function.

The CANFD module allows a maximum of 16 AFL entries.

During acceptance filtering process, each AFL entry in a channel is checked against the received message by the acceptance filter unit. The check starts from the lowest AFL entry number for this channel.

AFL search stops when a match of the received identifier with a configured identifier/mask combination occurs or when the received identifier has been compared against all AFL entries defined for the related channel. If no match occurs, then the received message is rejected. No notification is given to the application in this case.

Additionally, an automatic DLC filtering is performed for each accepted message if DLC check is globally enabled. If the DLC value of the received message is equal to or higher than the configured DLC value in the matching AFL entry, the DLC check is passed.

If DLC replacement (`CFDGCFCG.DRE` bit) is enabled, DLC value configured in the matching AFL entry is greater than 0x0 and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination RXMB or FIFO Buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received on the CAN Bus are not stored in the destination RXMB or FIFO Buffer. These additional data bytes are stored as 0x00 in the destination RXMB or FIFO Buffer.

If DLC replacement is enabled and DLC value of matching AFL entry is 0x0, then the received value of DLC is stored in the destination RX MB or FIFO Buffer.

If DLC replacement (the `CFDGCFCG.DRE` bit) is disabled and DLC check passes, then the received value of DLC on the CAN bus is stored in the destination RXMB or FIFO buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received from the CAN bus are also stored in the destination RXMB or FIFO buffer.

If DLC value of the received message is less than the configured DLC value in the matching AFL entry, then DLC check fails. In this case, the received message is rejected and is not stored in any RXMB or FIFO buffer.

Additionally, DLC check failure is flagged by the DLC Error Flag in the Global Error Flag Register. If configured, an error interrupt is also generated. The DLC replacement configuration has no impact if the DLC check fails.

If a message has passed both acceptance filtering and DLC filtering, it is stored in a single reception message buffer and/or in FIFO buffers configured for reception function.

This message storage target information is also defined in the same AFL entry. Do not set a target at the AFL entry which is not configured.

Each accepted received message can be stored into a maximum of 2 different target destinations (single reception message buffer and/or FIFO buffers).

The programming of more than 2 target destinations is not allowed. If more destinations are programmed, then the internal timing might lead to a race condition that prevents the storage of received messages in the message RAM.. Correct configuration of the numbers of target destination is the responsibility of the application.

Additional protection mechanism is made for the case when a received message contains more data payload Bytes than possible to store in the target destination (`CFDRMNB.RMPLS`, `CFDRFCCa.RFPLS` or `CFDCFCC.CFPLS`).

If `CFDGCFCG.CMPOC` = 0, the message is completely rejected and is stored in the target destination. When `CFDGCFCG.CMPOC` = 0 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (`CFDRMNB.RMPLS`, `CFDRFCCa.RFPLS` or `CFDCFCC.CFPLS`), the corresponding `CFDFMSTS.RFxMLT` or `CFDFMSTS.CFxMLT` bit is not set to 1, respectively.

When `CFDGCFG.CMPOC = 1`, the received data bytes greater than `CFDRMNB.RMPLS` is rejected. When `CFDGCFG.CMPOC = 1` and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (`CFDRMNB.RMPLS`, `CFDRFCCa.RFPLS` or `CFDCFCC.CFPLS`), the corresponding `CFDFMSTS.RFxMLT` or `CFDFMSTS.CFxMLT` bit is set to 1, respectively.

Depending on the `CFDGCFG.DRE` bit, the original received DLC or the DLC value configured at the AFL entry is stored.

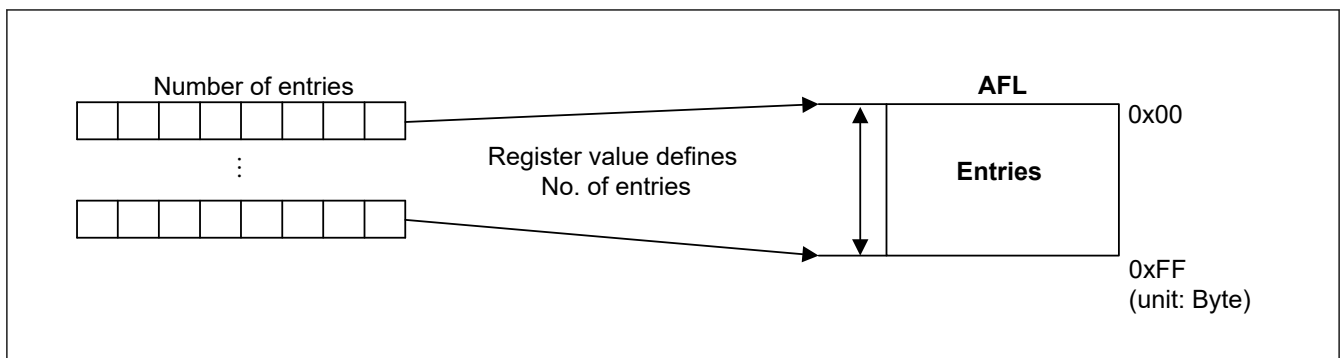
Regardless of the `CFDGCFG.CMPOC` bit setting, `CFDGERFL.CMPOF` is set to 1 if a payload overflow condition is detected.

The DLC filtering is performed before the payload overflow function. So for one reception frame, only one flag can be set at the same time with `CFDGERFL.DEF` or `CFDGERFL.CMPOF`\*1.

Note 1. This bit is not available in the classical CAN function.

### 32.5.2 Allocation of AFL Entries

The number of AFL entries per channel can be configured using the dedicated field in the related Global Acceptance Filter Configuration Registers (see [Figure 32.24](#)).



**Figure 32.24 Configuration of AFL for each channel**

The minimum number of entries for one channel is 0 (no entries defined for the channel) and the maximum number of entries is 16.

All entries are unique for a channel and overlapping or sharing of entries is not supported. Correct configuration of the AFL is the responsibility of the application.

The CANFD module does not flag errors related to the configuration of the AFL.

### 32.5.3 AFL Entry Description

Each AFL entry consists of 16 bytes. The fields in all entries are identical.

Each entry contains the following information for acceptance filtering and DLC filtering:

- Identifier (11 bits for Standard Frame format, 29 bits for Extended Frame format):  
Acceptance filter unit checks the identifier field of the received message against the identifier field of each AFL entry (full 29 bits masking of identifier bits is possible, see information that follows).
- IDE bit:  
Acceptance filter unit checks the IDE bit of the received message against this bit and selects the relevant part of the identifier field for acceptance filtering (masking of IDE bit is possible, see the information that follows).
- RTR bit:  
Acceptance filter unit only accepts data frames ( $RTR = 0$ ) or remote frames ( $RTR = 1$ ) according to the setting of this bit (masking of RTR bit is possible, see the information that follows).
- Loopback Configuration bit:  
This bit can enable or disable the AFL entry depending on the Loopback Configuration or Mirror mode condition.
- Mask for Identifier bits (29 bits):

Each bit in the identifier mask can mask the corresponding identifier bit in the AFL entry during acceptance filtering, see [Figure 32.25](#).

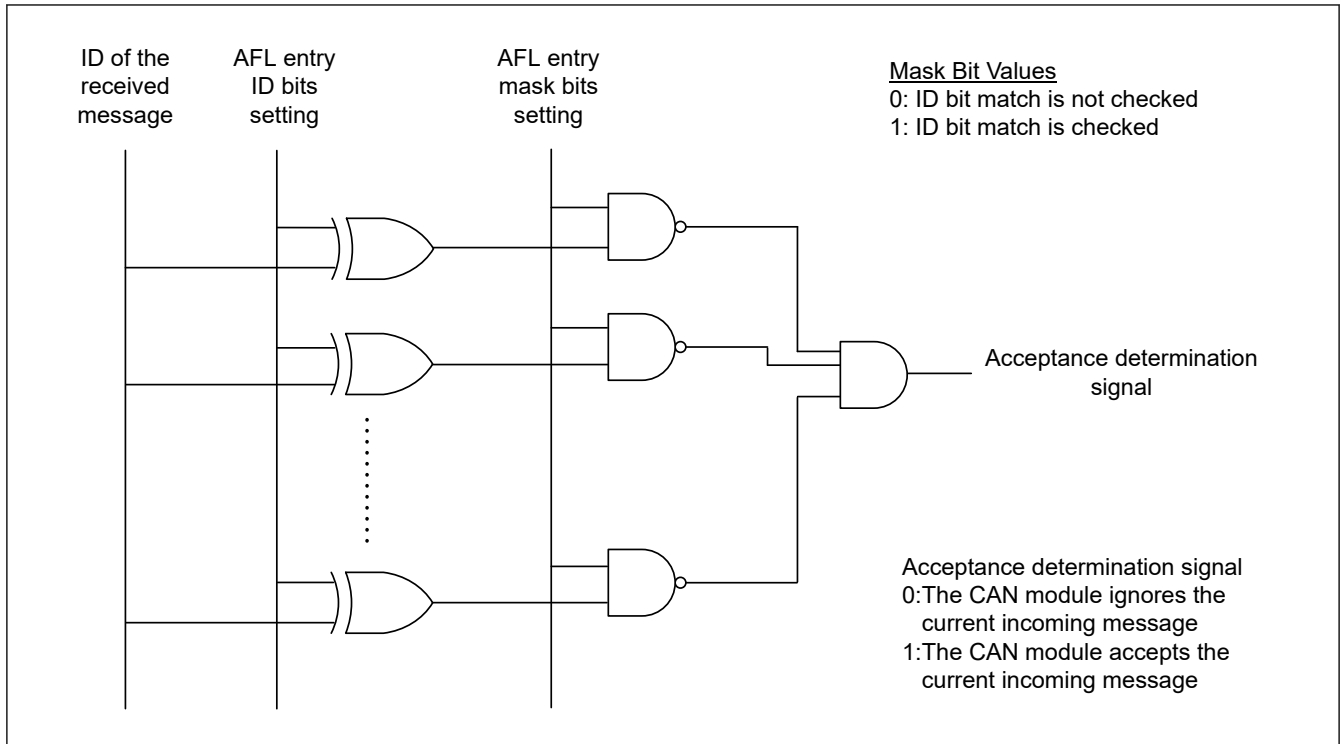
- **Mask for IDE bit:**  
If this Mask bit masks the IDE bit of the AFL entry in both Standard Identifier and Extended Identifier format, messages can be accepted by this AFL entry. The identifier of the received message is compared against the Standard Identifier part of the AFL entry for Standard Identifier format messages and against the Extended Identifier part of the AFL entry for Extended Identifier format messages.
- **Mask for RTR bit:**  
If this Mask bit masks the RTR bit of the AFL entry in both frame formats, data frame and remote frame formats are accepted by this AFL entry.
- **Pointer information (16 bits):**  
This 16-bit pointer is attached to a received message accepted by the related AFL entry. The pointer is added during message storage in the message buffer area and can be used by application as support function. The pointer information can be used for example to support PDU identifier allocation for the received message in AUTOSAR systems.
- **Information label (2 bits):**  
This 2-bit label is attached to a received message accepted by the related AFL entry. The label is added during message storage in the message buffer area and can be used by application as support function.
- **DLC value for automatic DLC filtering:**  
If the DLC value of the received message is equal or higher than the configured DLC value, the DLC check is passed.

If the DLC value in this AFL entry is configured to 0, DLC filtering is effectively disabled for this entry (all accepted messages pass DLC filtering).

Each AFL entry contains the following information for the handling of received messages:

- Message buffer number of one single reception message buffer as target for received message storage
- Single reception message buffer enable bit to configure the single reception message buffer number to be valid or invalid, as target for received message storage
- FIFO direction pointer - each bit of the FIFO direction pointer configures a dedicated FIFO as possible target for a received message

There is no hardware protection against such storage of message. Therefore, the FIFO direction pointer must be configured carefully.



**Figure 32.25** Acceptance function

### 32.5.4 Entering Entries in the AFL

Application software can enter one full entry into the AFL using the following registers:

- Global AFL ID Entry Register: Part 1 of the AFL entry
- Global AFL Mask Entry Register: Part 2 of the AFL entry
- Global AFL Pointer 0 Entry Register: Part 3 of the AFL entry
- Global AFL Pointer 1 Entry Register: Part 4 of the AFL entry.

16 sets of these registers form a group of AFL entries. The AFL should only be configured in CH\_RESET or CH\_HALT mode.

Follow the configuration shown in [Figure 32.26](#) to program the AFL.

After entering all entries in Configuration mode, locking of the AFL access should be performed to protect unwanted write access to the AFL.

Write protection is active during all Global modes (GL\_RESET, GL\_HALT, and GL\_OPERATION) if the lock bit is set.

Read access to AFL is still possible during all Global modes even when AFL data access is disabled (consistency check of AFL contents is possible during run time).

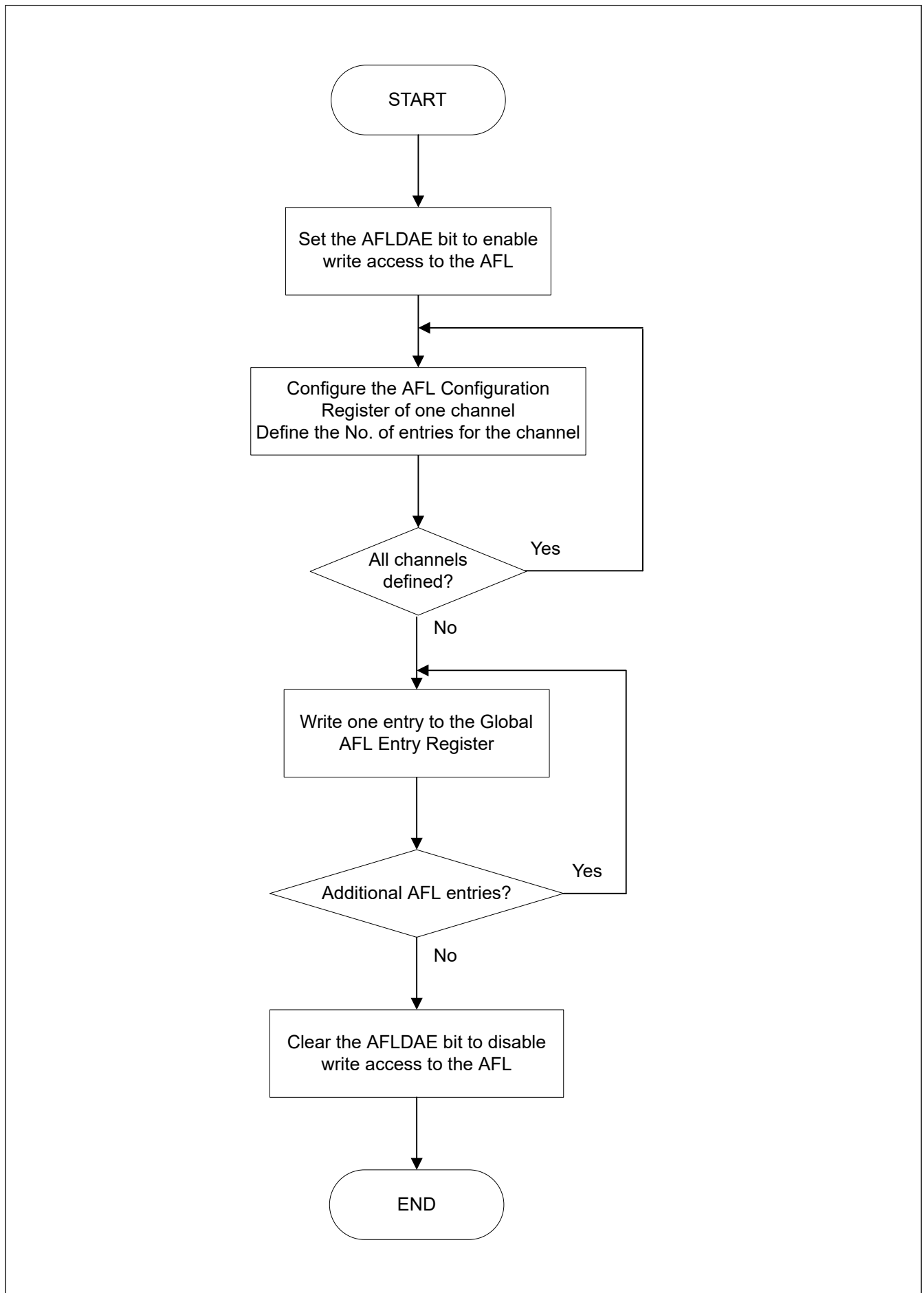


Figure 32.26 AFL configuration flow  
R01UH1129EJ0100 Rev.1.00  
Oct 23, 2024

### 32.5.5 Loopback Modes

If the Loopback Configuration bit is set, the AFL entry is only valid in Loopback test mode (Self-test mode 0 or Self-test mode 1) or in mirror mode when receiving messages that were transmitted by the respective CAN channel itself.

The AFL entry is not valid for received messages in loopback mode transmitted by other CAN nodes on the bus. The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID respectively.

If the Loopback Configuration bit is 0, the AFL entry is only valid for:

- Received messages transmitted by other CAN nodes on the bus in normal (non-loopback mode) and mirror modes
- Received messages transmitted by other CAN nodes or the CAN channel itself in Loopback test mode.

The mirror mode can be enabled with the CFDGCFG.MME bit in the Global Configuration Register. If CFDGCFG.MME bit is set, then a successfully transmitted message can be stored back in an RX message buffer or FIFO buffer if a matching entry is configured in the AFL for that channel.

The Loopback Configuration bit in the matching AFL entry must be set to store this frame.

If Mirror mode and Loopback test mode are configured at the same time, the Loopback test mode behavior applies.

Table 32.22 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

**Table 32.22 Behavior of acceptance filter based on the loopback configuration setting in AFL entry**

Mirror Mode Enable (MME Configuration bit)	Loopback in test mode (Self-test mode 0 or Self-test mode 1)	Channel mode	Loopback Configuration bit in AFL entry	AFL entry
0	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Invalid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid
1	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Valid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid

Note: The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID, respectively.

### 32.5.6 IDE Masking

When the GAFLIDEM bit is 0 in an AFL entry, the IDE bit configured in the AFL entry is not used for ID matching. In this case, the use of ID[10:0] or ID[28:0] matching is based on the received IDE bit.

Consider the following example:

- The ID and Mask fields of an AFL entry x is configured as follows:
  - CFDGAFLLID [x] = 0xC0553A20 → IDE = 1, RTR = 1, LLB = 0, ID[10:0] = 0x220 / ID[28:0] = 0x00553A20
  - CFDGAFLLMr = 0x0000FFFF → IDEM = 0, RTRM = 0, IDM[10:0] = 0x7FF / IDM[28:0] = 0x0000FFFF



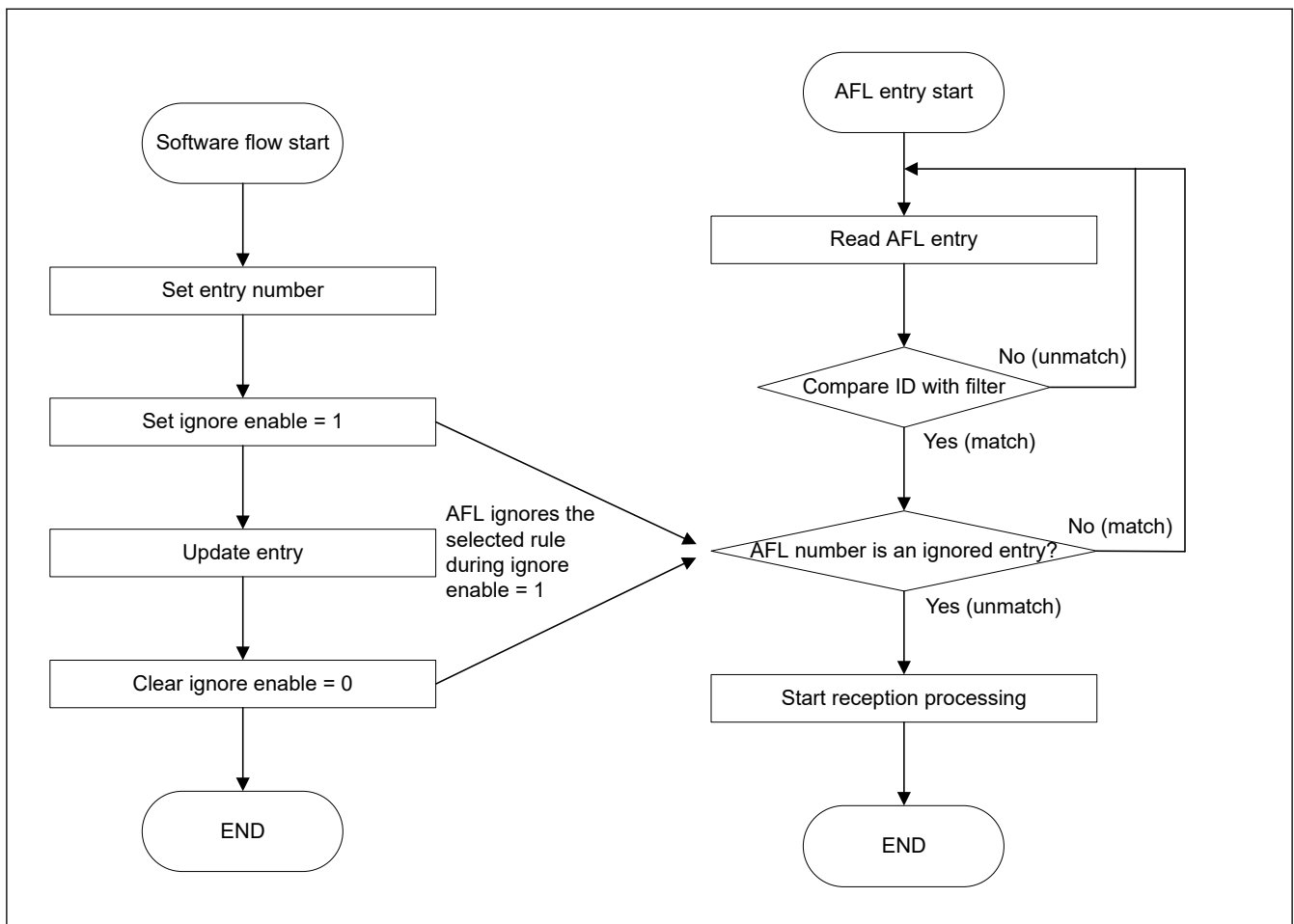
- The comparison result for the four different received IDs with AFL entry x is described as follows:
  - If a frame with IDE = 0 and ID = 0x220 is received, this is considered as a match
  - If a frame with IDE = 0 and ID = 0x320 is received, this is not a match
  - If a frame with IDE = 1 and ID = 0x1FFF3A20 is received, this is considered as a match
  - If a frame with IDE = 1 and ID = 0x08803220 is received, this is not a match.

### 32.5.7 Updating AFL Entry during Communication

You can update the AFL entry without disabling all CAN communications. Choose the entry number to be updated by setting the AFL entry number, and ignore the enable bit.

This entry number is ignored from the AFL matching while the entry is being updated.

Figure 32.27 shows the update flow for an AFL entry.



**Figure 32.27 Update flow for an AFL entry**

The method to update an AFL entry is as follows:

1. Set the entry number to CFDGAFALIGNENT register.
2. Set the value 0xC401 (key code and enable bit) to CFDGAFALIGNCTR register.
3. CFDGAFLECTR.AFLDAE is set to 1.
4. Set the new rule to CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r registers.
5. CFDGAFLECTR.AFLDAE is cleared to 0.
6. Set the value 0xC400 (key code and clear enable bit) to CFDGAFALIGNCTR register.

**Note:** This entry number is ignored during the periods from (2) to (5).

(1) Example 1: Deleting an entry

Deleting entry3 when the total number of entries is 6 channel.

total entry = 6	entry0	0	ID = 0x050	
	entry1	1	ID = 0x051	
	entry2	2	ID = 0x052	
	entry3	3	ID = 0x053	← delete rule
	entry4	4	ID = 0x054	
	entry5	5	ID = 0x055	

**How to delete an entry**

1. Set 0x00000003 to CFDGAFLIGNENT register.
2. Set 0x0000C401 to CFDGAFLIGNCTR register.
3. Set 0x00000100 to CFDGAFLECTR register.
4. Set the same rule as the previous rule by accessing CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r (r = 3, this is entry3).
5. Set 0x00000000 to CFDGAFLECTR register.
6. Set 0x0000C400 to CFDGAFLIGNCTR register.

Entry3 is now deleted.

total entry = 5 entry2 = entry3	entry0	0	ID = 0x050	
	entry1	1	ID = 0x051	
	entry2	2	ID = 0x052	
	entry3	3	ID = 0x052	← set the same rule as the previous rule
	entry4	4	ID = 0x054	
	entry5	5	ID = 0x055	

(2) Example 2: Adding an entry

Adding a new entry to entry3 when the total number of entries is 6.

total entry = 5 entry2 = entry3	entry0	0	ID = 0x050	← add new rule in this position
	entry1	1	ID = 0x051	
	entry2	2	ID = 0x052	
	entry3	3	ID = 0x052	
	entry4	4	ID = 0x054	
	entry5	5	ID = 0x055	

### How to add an entry

1. Set 0x00000003 to CFDGAFLIGNENT register.
2. Set 0x0000C401 to CFDGAFLIGNCTR register.
3. Set 0x00000100 to CFDGAFLECTR register.
4. Set the new rule by accessing CFDGAFLIDr, CFDGAFLMr, CFDGAFLP0r, CFDGAFLP1r (r = 3, this is entry3).
5. Set 0x00000000 to CFDGAFLECTR register.
6. Set 0x0000C400 to CFDGAFLIGNCTR register.

The new entry is now added.

total entry = 6	entry0	0	ID = 0x050	← add new rule
	entry1	1	ID = 0x051	
	entry2	2	ID = 0x052	
	entry3	3	ID = 0x056	
	entry4	4	ID = 0x054	
	entry5	5	ID = 0x055	

The AFL filter can be used to set CFDGAFLCFG, and addition/deletion of an entry is possible. Therefore, it is necessary to set the maximum number to be used to CFDGAFLCFG.

## 32.6 FIFO Buffers and Normal Message Buffer Configuration

This section describes the process for configuring the number of RX message buffers, the FIFO buffers, and the flat TX message buffers in the CANFD module. The message buffers are mapped as shown in [Figure 32.28](#).

The RX message buffers can be accessed with the RX Message Buffer Registers.

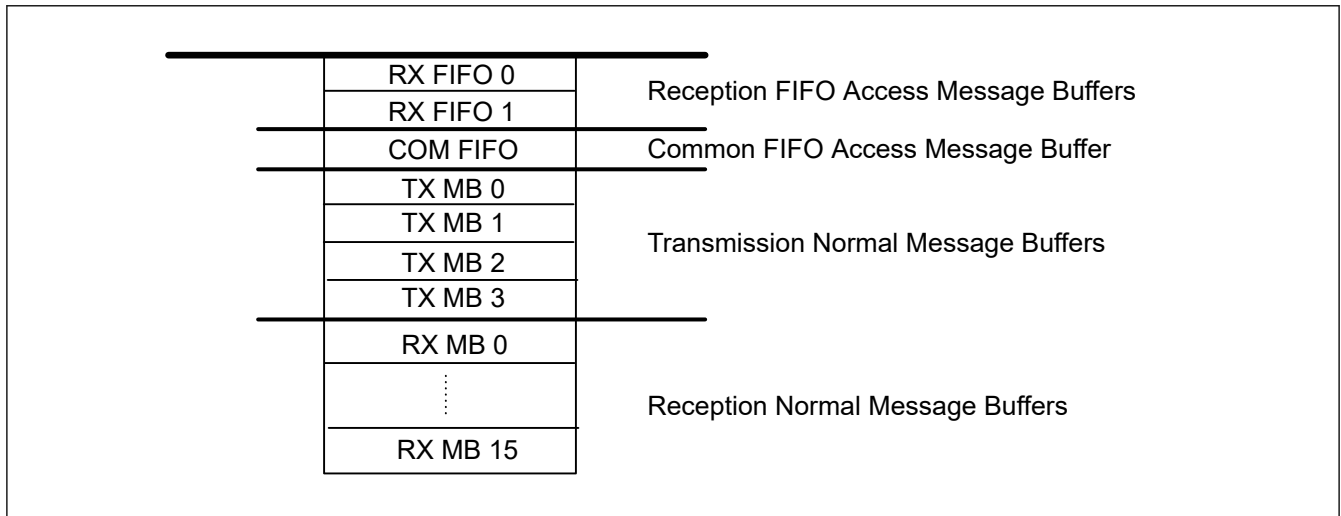
The RX FIFO buffers and the common FIFO buffers configured in RX mode or TX mode can only be accessed with the FIFO Access Registers.

If the common FIFO is configured in TX mode, you can only write data into the FIFO buffer using the FIFO Access registers.

If the common FIFO is configured in RX mode, you can only read data from the FIFO Access Registers.

The TX message buffers can be accessed with the TX Message Buffer Registers.

If unused message buffer locations are read, the message buffer locations are read as unknown values.



**Figure 32.28** Message buffer configuration

### 32.6.1 Normal RX Message Buffers

In CANFD module, the frames received can be stored in normal RX message buffers based on the configuration of the AFL entries.

Additionally, the number of normal RX message buffers required in the system can be chosen up to a fixed maximum limit.

#### 32.6.1.1 Normal RX Message Buffer Configuration

In CANFD module, the number of normal RX message buffers can be configured by writing to the RX Message Buffer Number Register.

The limiting values for the configuration of number of message buffers are:

- Minimum value = 0x00 (no normal RX MB)
- Maximum value = 0x10

Do not use values outside these limits.

The AFL entries for routing the received messages to normal RX message buffers must be configured to match the requirements of the system.

The AFL entries must also be configured properly, and an AFL entry for normal RX message buffers should not exceed the number of message buffers configured in the RX Message Buffer Number Register.

**Note:** There is no internal check procedure provided in CANFD module against wrong configuration of the AFL.

The data field size of the RX message buffer can be configured with the CFDRMNB.RMPLS bit. The default size is 8 bytes and the maximum data payload size is 64 bytes.

When the receiving frame exceeds the data field size, then the acceptance depends on the configuration of CFDGCFG.CMPOC (message rejecting or data payload cut).

**Note:** RMPLS and CMPOC bit is not available in the classical CAN function, so, these feature is not valid for classical CAN.

### 32.6.2 FIFO Buffers

The CANFD module provides a fixed number of FIFO buffers to support storage of frames for reception and transmission functions.

The number of reception-only FIFO buffers is fixed to 2. However, common FIFO buffer channel can be configured to store messages for transmission or reception function.

These FIFO buffers can be enabled or disabled, and the following parameters can be configured to match the system requirements:

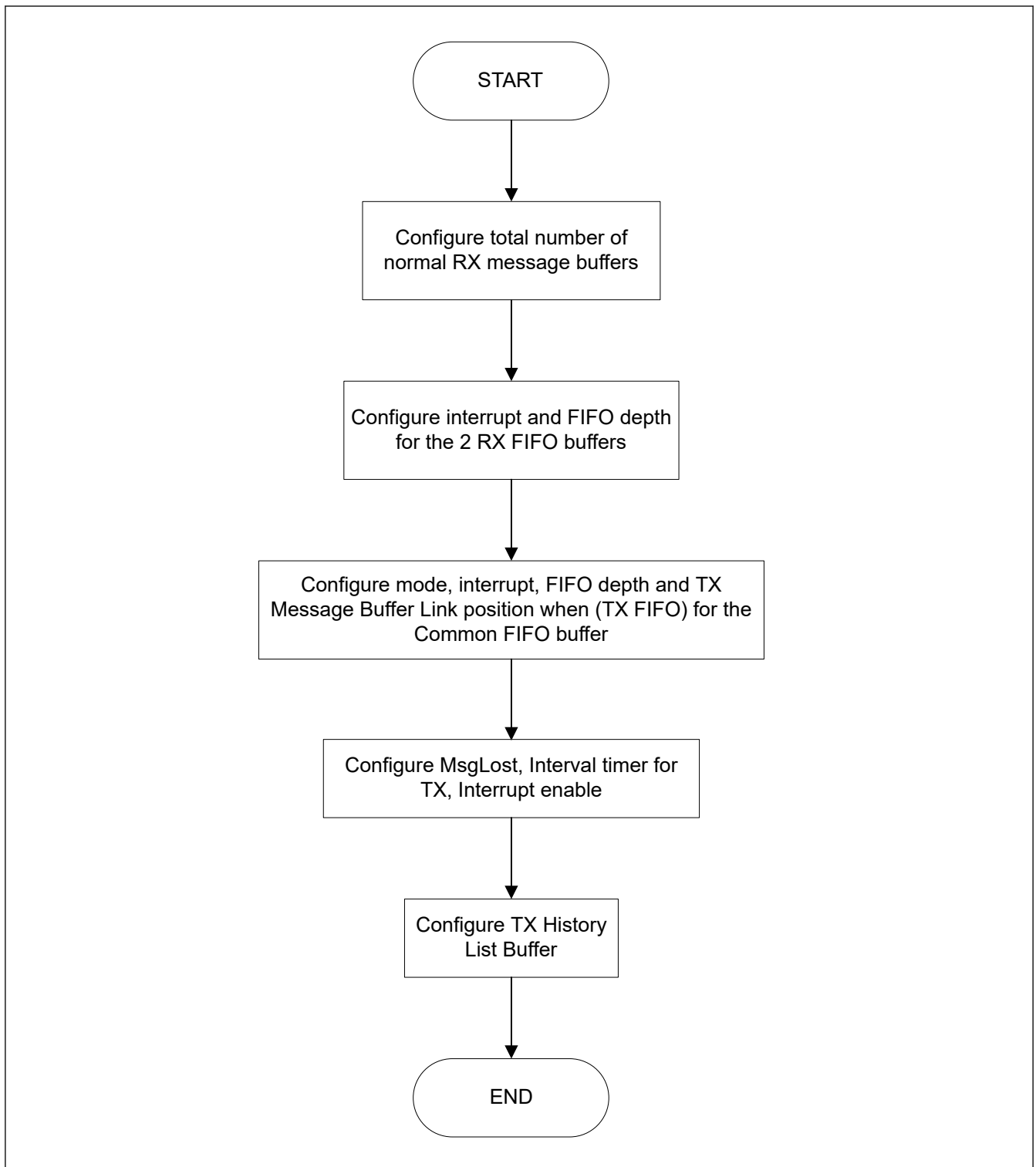
- Size
- Interrupt structure
- Message lost mechanism
- Message overwrite mechanism of the FIFO buffers
- Location of the TX FIFO.

When the receiving frame exceeds the data field size, the acceptance depends on the configuration of the `CFDGCFG.CMPOC` bit (message rejecting or data payload cut).

### 32.6.2.1 FIFO Buffers Configuration

In CANFD module, the FIFO buffers can be configured to match the system requirements.

The total number of FIFO buffers = 2 RX FIFO buffers + 1 common FIFO buffer = 3 FIFO buffers.



**Figure 32.29** FIFO buffer configuration flow in CANFD module

As shown in [Figure 32.29](#), the various FIFO buffers can be configured by writing to the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

For the 2 RX FIFO buffers, the following parameters can be configured:

- Interrupts
- FIFO depth
- FIFO payload data size.

For the common FIFO buffer, the following parameters can be configured:

- Mode
- Interrupts FIFO depth
- FIFO payload data size
- FIFO TX link position.

### (1) FIFO mode configuration of Common FIFO buffer

The mode of the common FIFO buffer can be configured by writing to the CFDCFCC.CFM[1:0] bits in the Common FIFO Configuration/Control Register. The possible modes of configuration for Common FIFO buffer are:

- 0b RX mode (default mode after hardware reset)
- 1b TX mode

Messages can only be read from the RX FIFO buffers and the Common FIFO buffer configured in RX mode. Messages are stored by the CAN module in these FIFO buffers based on the AFL entries.

Messages can be read and written into the Common FIFO buffer configured in TX mode. These messages are transmitted on the appropriate CAN channel.

The pointers can only be incremented when a new message is stored in the FIFO buffer and decremented when a message is transmitted on the corresponding CAN channel by the CANFD module.

After a hardware reset, the Common FIFO buffer is configured in RX mode by default. Only enable the FIFO buffers after configuring the Common FIFO buffer in the required modes.

### (2) FIFO TX message buffer link configuration

When the common FIFO is configured as TX FIFO, the FIFO buffer must be linked to a normal TX message buffer to participate in the transmission scan.

Do not write data into a TX message buffer that is linked to a Common FIFO buffer. Also, the TX message buffer linked to a Common FIFO buffer should not be a part of the TX Queue.

The TX message buffer link of each Common FIFO buffer can be configured by writing to the CFDCFCC.CFTML[1:0] bits in the Common FIFO Configuration/Control Registers. Available options for TX message buffer link configuration are:

- 0x00: TX Message Buffer 0
- 0x01: TX Message Buffer 1
- 0x10: TX Message Buffer 2
- 0x11: TX Message Buffer 3

### (3) FIFO depth configuration

The depth of each FIFO buffer can be configured by writing to the CFDRFCCa.RFDC[2:0] bits and CFDCFCC.CFDC[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The 6 available options for depth configuration are:

- 0x000: 0 Message (FIFO buffer cannot be enabled)
- 0x001: 4 Messages
- 0x010: 8 Messages
- 0x011: 16 Messages
- 0x100: 32 Messages
- 0x101: 48 Messages

The RAM allocation for RX message buffers along with FIFO buffers is limited to 16 messages with 64 data bytes (76 bytes including ID and PTR). Configuration of the RX message buffers, along with FIFO buffers, that exceeds this maximum limit should not be done.

CANFD module logic does not check the validity of the configuration.

Note: If the FIFO depth of a common FIFO is 4 messages or more (CFDCFCC.CFDC[2:0] > 000b), then the Common FIFO TX message buffer link is valid when the FIFO is disabled or enabled.  
If FIFO depth is 0 messages, then the Common FIFO TX message buffer link is not valid when the FIFO is disabled or enabled.

#### (4) FIFO payload size configuration

The data size of each FIFO buffer can be configured by writing to the CFDRFCCa.RFPLS[2:0] bits and CFDCFCC.CFPLS[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The eight available options for depth configuration are:

- 000b: 8 bytes
- 001b: 12 bytes
- 010b: 16 bytes
- 011b: 20 bytes
- 100b: 24 bytes
- 101b: 32 bytes
- 110b: 48 bytes
- 111b: 64 bytes

The RAM allocation for RX message buffers along with FIFO buffers is limited to 16 messages with 64 data bytes (76 bytes including ID and PTR). Configuration of the RX message buffers, along with FIFO buffers, that exceeds this maximum limit should not be done.

CANFD module logic does not check the validity of the configuration.

Note: This feature is not available in the classical CAN function.

#### (5) FIFO interrupt configuration

The Interrupt generation conditions for the FIFO buffers can be configured by writing to the CFDRFCCa.RFIM and CFDCFCC.CFIM bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The two available options are:

- 0:
  - RX FIFO mode: Interrupt generated when the Common FIFO counter reaches CFDRFCCa.RFIGCV/CFDCFCC.CFIGCV value
  - TX FIFO mode: Interrupt generated when the Common FIFO transmits the last message successfully
- 1:
  - RX FIFO mode: Interrupt generated at the end of storage of every received message
  - TX FIFO mode: Interrupt generated for every successfully transmitted message

If the Interrupt Mode bit is 0 for a RX FIFO, then interrupt is generated based on the configuration of the CFDRFCCa.RFIGCV[2:0] bits.

Similarly, if the Interrupt Mode bit is 0 for a Common FIFO configured in RX mode, then interrupt is generated based on the configuration of CFDCFCC.CFIGCV[2:0] bits.

The eight available options for configuring the FIFO counter value for generation of an interrupt are:

- 000b: Interrupt generated when FIFO is 1/8th Full
- 001b: Interrupt generated when FIFO is 1/4th Full
- 010b: Interrupt generated when FIFO is 3/8th Full
- 011b: Interrupt generated when FIFO is 1/2 Full
- 100b: Interrupt generated when FIFO is 5/8th Full
- 101b: Interrupt generated when FIFO is 3/4th Full



- 110b: Interrupt generated when FIFO is 7/8th Full
- 111b: Interrupt generated when FIFO is Full.

In this case, an interrupt is generated when the message count matches the configured value.

However, there are some limitations on the configuration of the CFDRFCCa.RFIGCV[2:0] and CFDCFCC.CFIGCV[2:0] bits depending on the FDC[2:0] bits (FIFO Depth Configuration), see [Table 32.23](#).

**Table 32.23 FIFO interrupt generation counter and FIFO depth configuration**

RFDC[2:0] (CFDC[2:0])	RFIGCV[2:0] (CFIGCV[2:0])							
	111b	110b	101b	100b	011b	010b	001b	000b
000b	Don't care (FIFO cannot be enabled)							
001b	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed
010b	Allowed							
011b	Allowed							
100b	Allowed							
101b	Allowed							
110b	Allowed							
111b	Allowed							

### 32.6.2.2 FIFO Buffers Control

The FIFO interrupt must be enabled by setting any one of the following bits in the RX FIFO Configuration/Control Registers:

- CFDRFCCa.RFIE

In addition, the FIFO interrupt must be enabled by setting any one of the following bits in the Common FIFO Configuration/Control Register:

- CFDCFCC.CFRXIE
- CFDCFCC.CFTXIE

After configuration is complete, each FIFO can be enabled by setting the CFDRFCCa.RFE and CFDCFCC.CFE bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Register to allow transmission and reception of messages.

## 32.7 Interrupts and DMA

### 32.7.1 Interrupts

The CANFD module generates several interrupts. The interrupt output, which is connected to the Interrupt Controller Unit (ICU), can be controlled by the corresponding interrupt enable bit.

The status flag is set independent from this enable bit.

The channel transmission interrupt has an additional status flag register. The status bits are set when the corresponding interrupt enables are set.

The status flag register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.

The interrupts in the CANFD module can be classified into two groups, global interrupts and channel interrupts:

- Global interrupts:  
The CANFD module can generate 3 global interrupts:
  - Global interrupt for successful reception into the 2 RX FIFO buffers
  - Global error interrupt.
  - Global Interrupt for successful reception into the 16 RX message buffers

- Channel interrupts:  
Channel of the CANFD module can generate 3 channel interrupts:
  1. Channel transmission
    - Transmission completion from channel
    - Transmission abort from channel
    - Transmission from TX Queue for a channel
    - Channel THL interrupt
    - Successful transmission from a Common FIFO in TX mode for a channel.
  2. Channel error interrupt
  3. Successful reception in a Common FIFO in RX mode for a channel.

The interrupts are cleared when the corresponding flag bits are cleared or the Interrupt enable bits are cleared.

Table 32.24 gives an overview of interrupt sources for the different interrupt outputs. The interrupt outputs are active-high.

**Table 32.24** Interrupt source overview

Parameter	Interrupt	Name	Interrupt source	Interrupt clearing
Global Interrupts	Successful reception into at least one RX FIFO	CAN_RXF	Interrupt flag of corresponding RX FIFO for which interrupt is enabled	Clear the interrupt flag of corresponding RX FIFO buffer for which interrupt is enabled
	Global Error	CAN_GLERR	Any of the following: <ul style="list-style-type: none"> <li>● DLC Error flag</li> <li>● Message Lost Status bit</li> <li>● TX History Entry Lost Status bit</li> <li>● CANFD Message Payload overflow flag</li> </ul>	Clear all of : <ul style="list-style-type: none"> <li>● DLC Error flag</li> <li>● Message Lost flags in all of the FIFO Status Registers</li> <li>● TX History List Entry Lost flag</li> <li>● CANFD Message Payload overflow flag</li> </ul>
	Successful reception into at least one RXMB	CANn_RXMB (n = 0, 1)	Interrupt flag of corresponding RXMB for which interrupt is enabled	Clear the interrupt flag of corresponding RXMB buffer for which interrupt is enabled
Channel Transmission Interrupts	Channel successful transmission	CANn_TX (n = 0, 1)	Any channel related TXMB Successful flag when interrupt is enabled*1	Clear all channel related TXMB Result status bits for which the interrupt is enabled
	Channel Abort		Any channel related TXMB Abort flag when interrupt is enabled*1	Clear all channel related TXMB Result Status bits for which the interrupt is enabled globally
	Channel transmission from TX Queue		Related channel TX Queue Interrupt flag	Clear related channel TX Queue Interrupt flag
	Channel THL Interrupt		Channel THL Interrupt status flag	Clear the relevant THL Interrupt status flag
	Channel COM FIFO TX Interrupt		Interrupt Flag for Common FIFOs in TX mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in TX mode belonging to the related channel
Channel Error Interrupt	Channel Error	CANn_CHERR (n = 0, 1)	Any channel related error flag in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register	Clear all channel related error flags in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register
Channel COM RX FIFO Interrupt	Channel COM FIFO RX Interrupt	CANn_COMFRX (n = 0, 1)	Interrupt flag for Common FIFOs in RX mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in RX mode belonging to the related channel

Note 1. These interrupts are only set for TX Message Buffers that do not belong to an enabled TX Queue and are not pointing to a common FIFO.  
Separate interrupts are provided for common FIFO buffers and TX Queue.

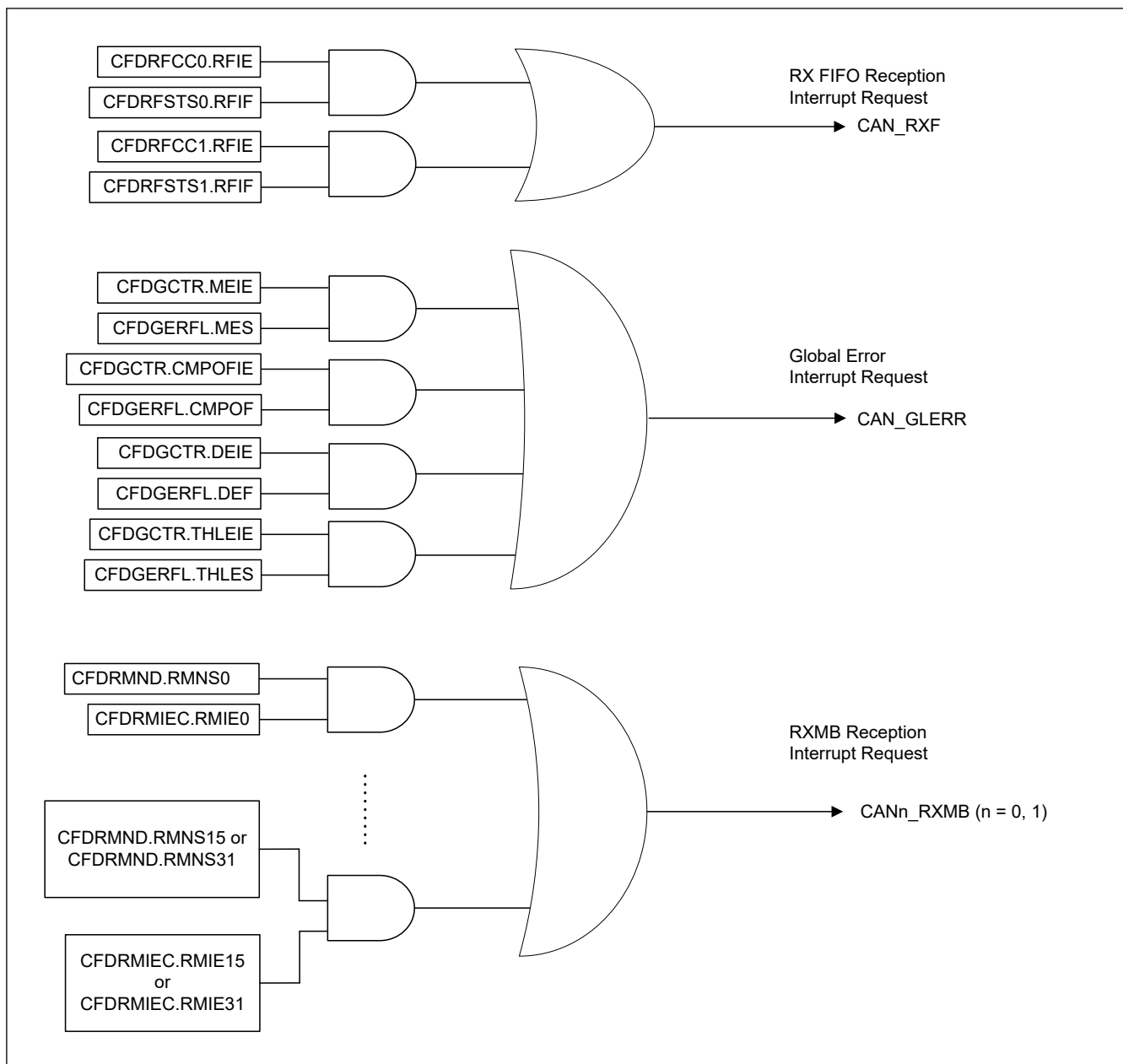


Figure 32.30 Global interrupt block diagram

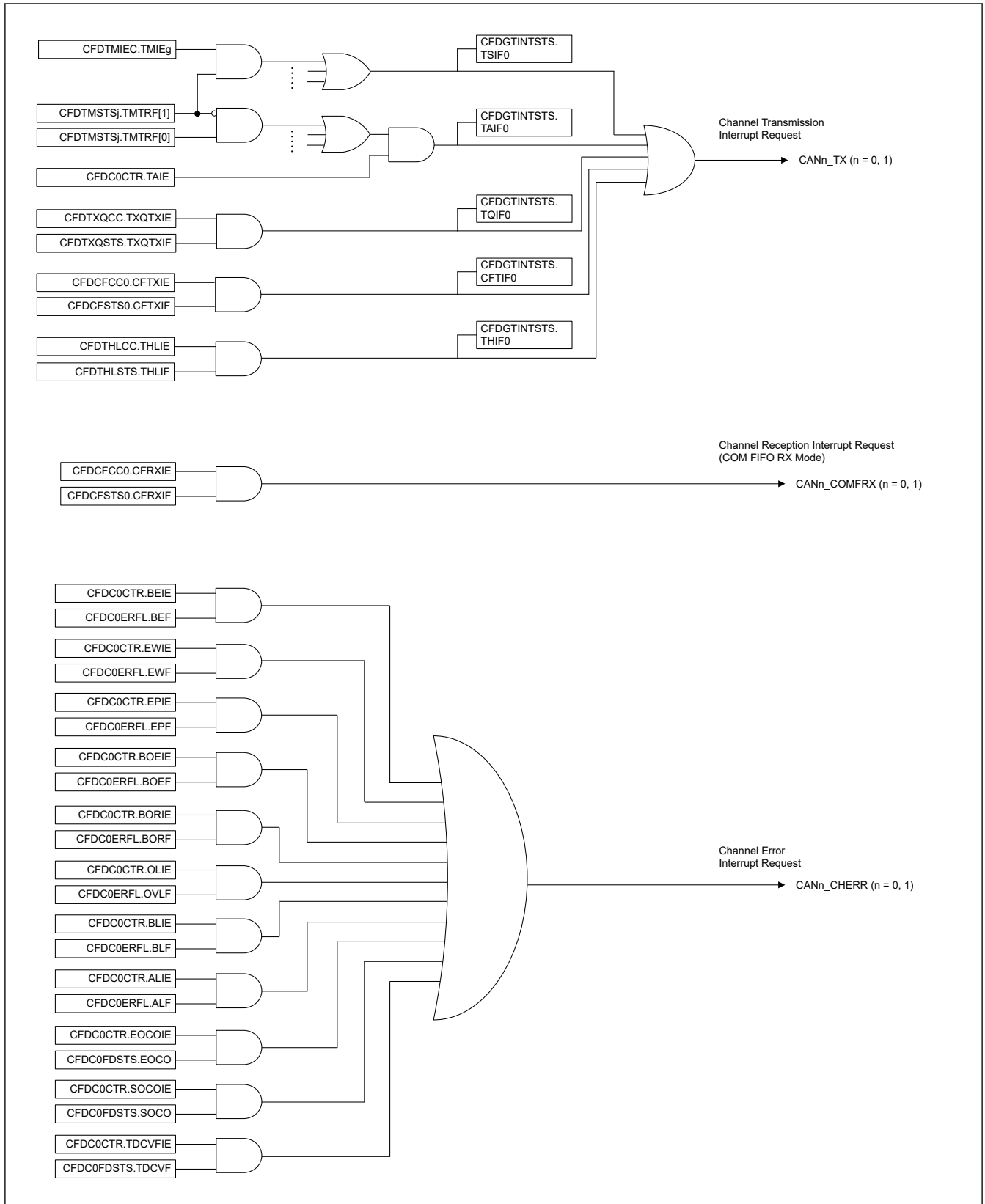


Figure 32.31 Channel interrupt block diagram

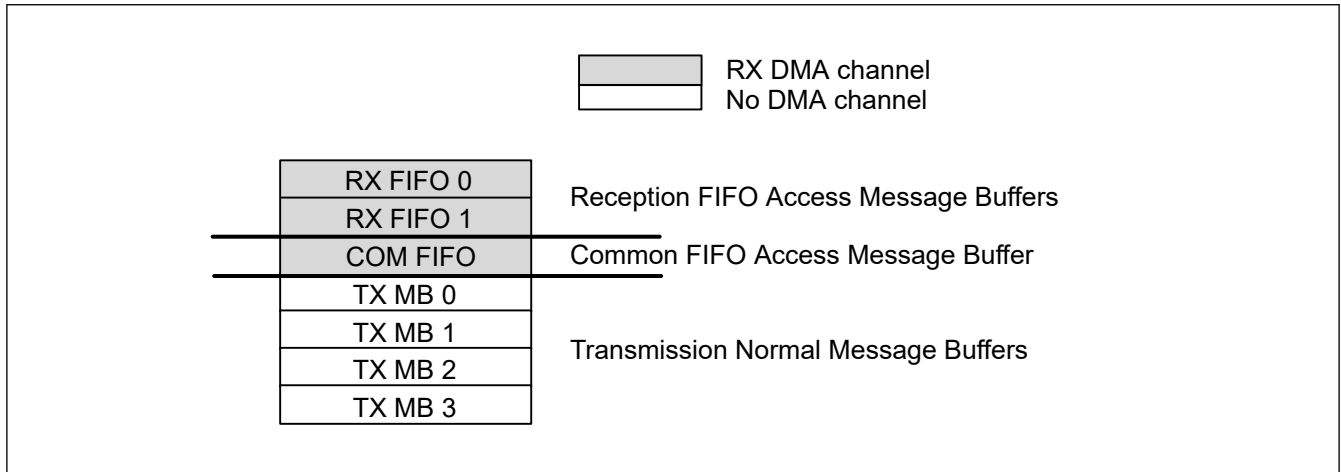
### 32.7.2 DMA Transfer

The CANFD module has message buffers that can be associated with a DMA channel:

- Reception DMA

- 2 RX FIFO message buffers
- Common FIFO Message Buffer

Figure 32.32 shows the potential DMA channels.



**Figure 32.32 Message buffer connectable to a DMA channel**

A DMA channel transfer request is generated for each FIFO entry to the DMAC when the related CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE is set to 1 and the belonging FIFO is not empty.

Reception FIFO Interrupt should be disabled for this particular FIFO (CFDRFCCa.RFIE or CFDCFCC.CFRXIE)

Use the regular start address for the DMA access window address. See Figure 32.33.

**Table 32.25 DMA channel access window address**

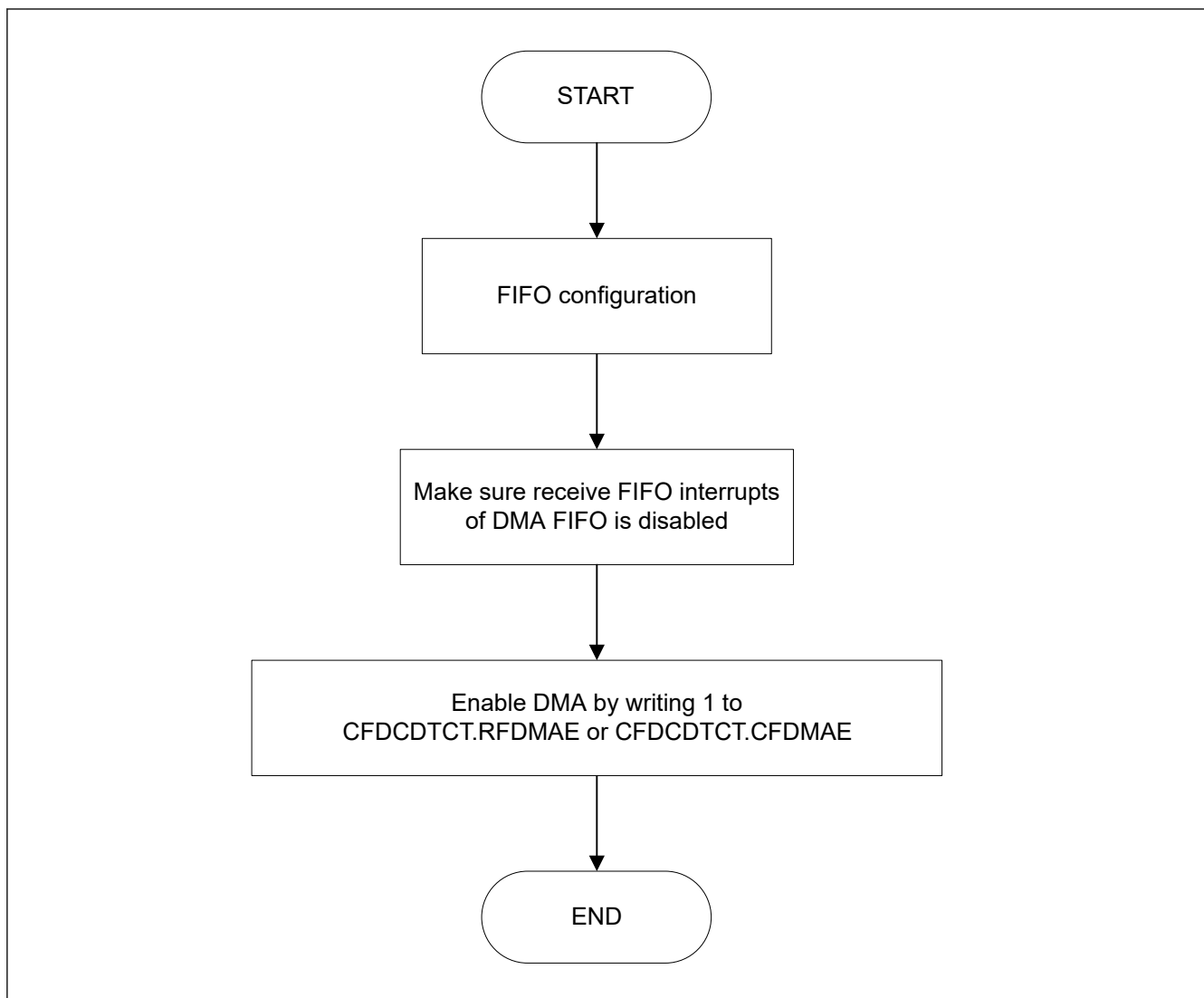
b = Message buffer component index	Message Buffer Component	Register	P	Regular Start Address
b = [0...1]	RFMBCPb[0]	CFDRFIDb	x	0x0520 + b × 0x004C
		CFDRFPTRb	x	0x0524 + b × 0x004C
		CFDRFFDSTsb	x	0x0528 + b × 0x004C
		CFDRFDFbp	[0...15]	0x052C + p × 0x0004 + b × 0x004C
—	CFMBCP0[0]	CFDCFID	x	0x05B8
		CFDCFPTR	x	0x05BC
		CFDCFFDCSTS	x	0x05C0
		CFDCFDFp	[0...15]	0x05C4 + p × 0x0004

DMA FIFO pointer decrement is done automatically by reading the last configured data payload byte (CFDRFCCa.RFPLS or CFDCFCC.CFPLS).

Note: The DMA must read the exact length of the configured data payload size (CFDRFCCa.RFPLS or CFDCFCC.CFPLS).

Note: This feature is not available for classical CAN function because CFDRFCCa.RFPLS and CFDCFCC.CFPLS are not in classical CAN.

Do not write to the FIFO control registers when DMA is enabled. The DMA enable of the particular DMA FIFO (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE) can be set at any time. Figure 32.33 shows a configuration flow for an initial setup.



**Figure 32.33 DMA enable flow**

To disable a DMA transfer request, you must disable the particular DMA enable bit (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE). If the disable is made during an ongoing transfer, then the transfer must be completed first before further action can be taken. The transfer status can be identified by the CFDCDTSTS.RFDMASTS or CFDCDTSTS.CFDMASTS bit. See [Figure 32.34](#) for the DMA disable flow. When the DMA is disabled, consideration should be made for the remaining or new incoming messages to this particular reception FIFO.

When the FIFO is not disabled, reception to the FIFO continues.

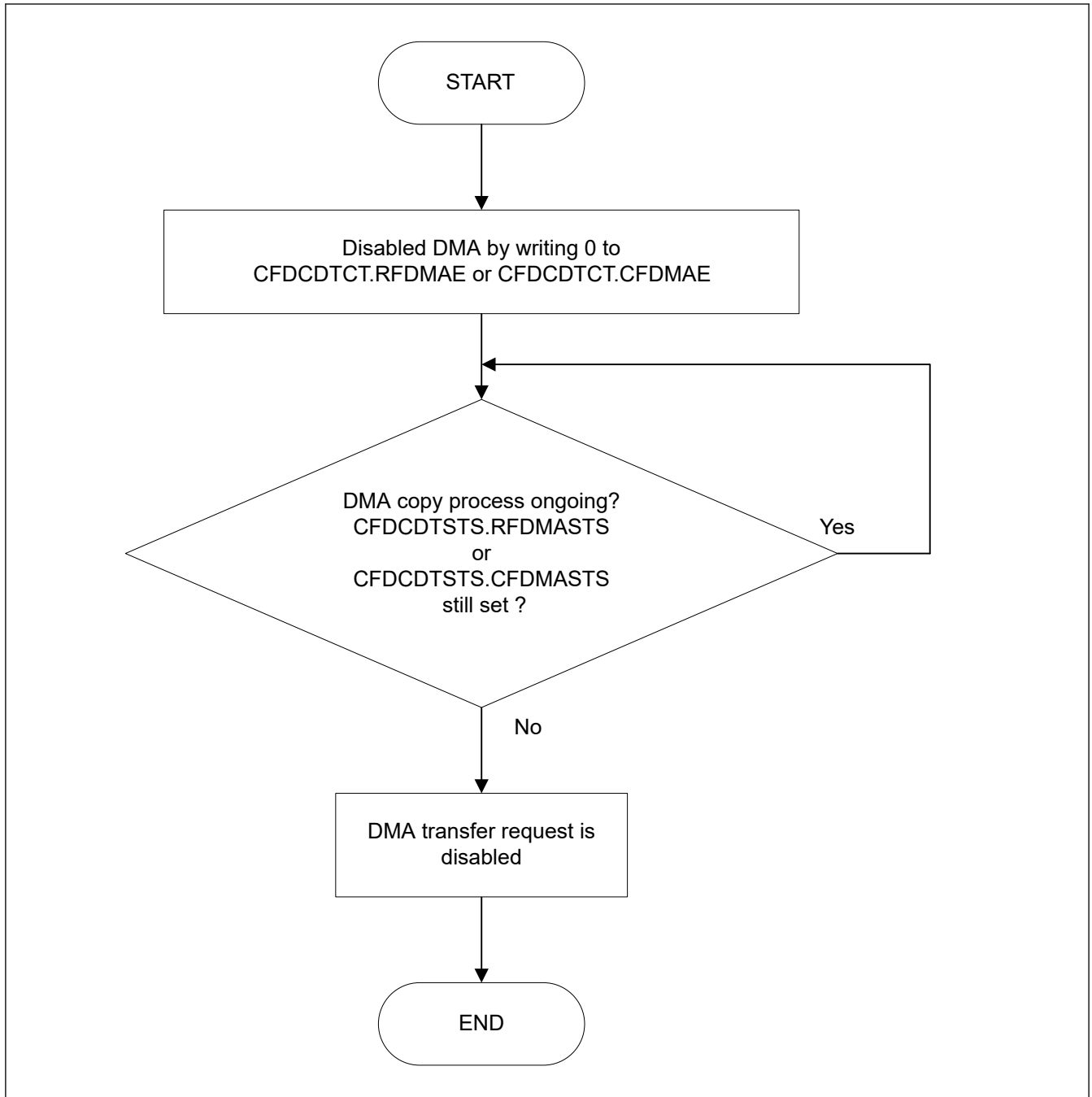


Figure 32.34 DMA disable flow

## 32.8 Reception and Transmission

### 32.8.1 Reception

In the CANFD module, CAN messages received on any of the channels are stored in RX message buffers, RX FIFO buffers, or Common FIFO buffers configured in RX mode depending on the Acceptance Filter List entries.

- Up to 16 RX message buffers can be configured
- 2 RX FIFO buffers available
- 1 Common FIFO Buffer can be configured in RX mode

### 32.8.1.1 Message Storage in RX Message Buffers

When a message is successfully received and stored in a RX message buffer, the corresponding New Data flag is set in the RX Message Buffer New Data Register.

The CAN message can be read from the corresponding RX message buffer.

If a new message is stored into a RX message buffer before the previous message in this message buffer can be read, then the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the RX message buffer. If such a loss of messages is not acceptable, then RX FIFO should be used for storing related messages.

Note: Users should do the same processing as the existing software flow also when using interrupt. (see [Figure 32.36](#))

Note: Unused data bytes are filled with 0x00 depending on the DLC value.

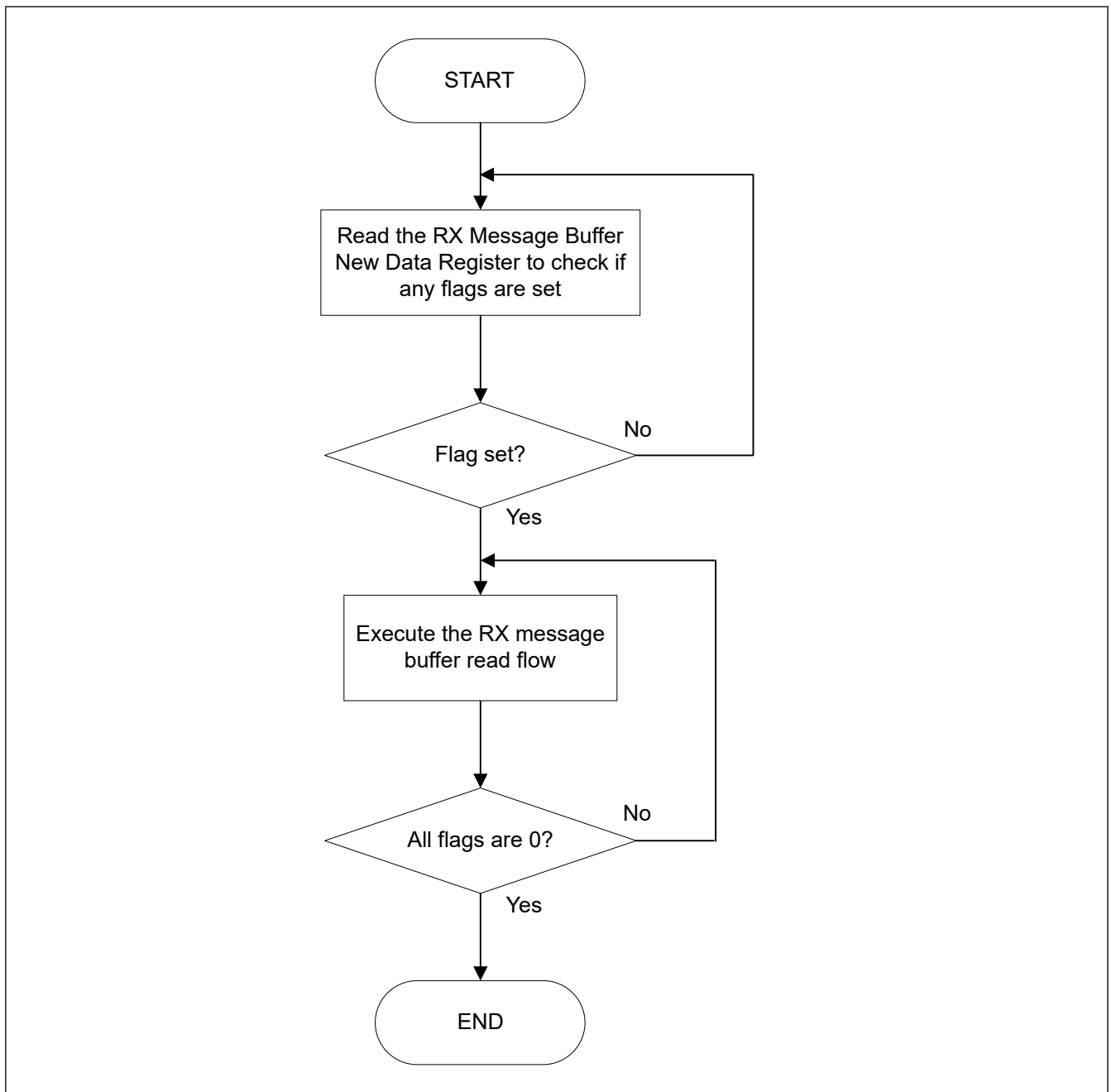


Figure 32.35 Access flow of RX message buffer (Polling)



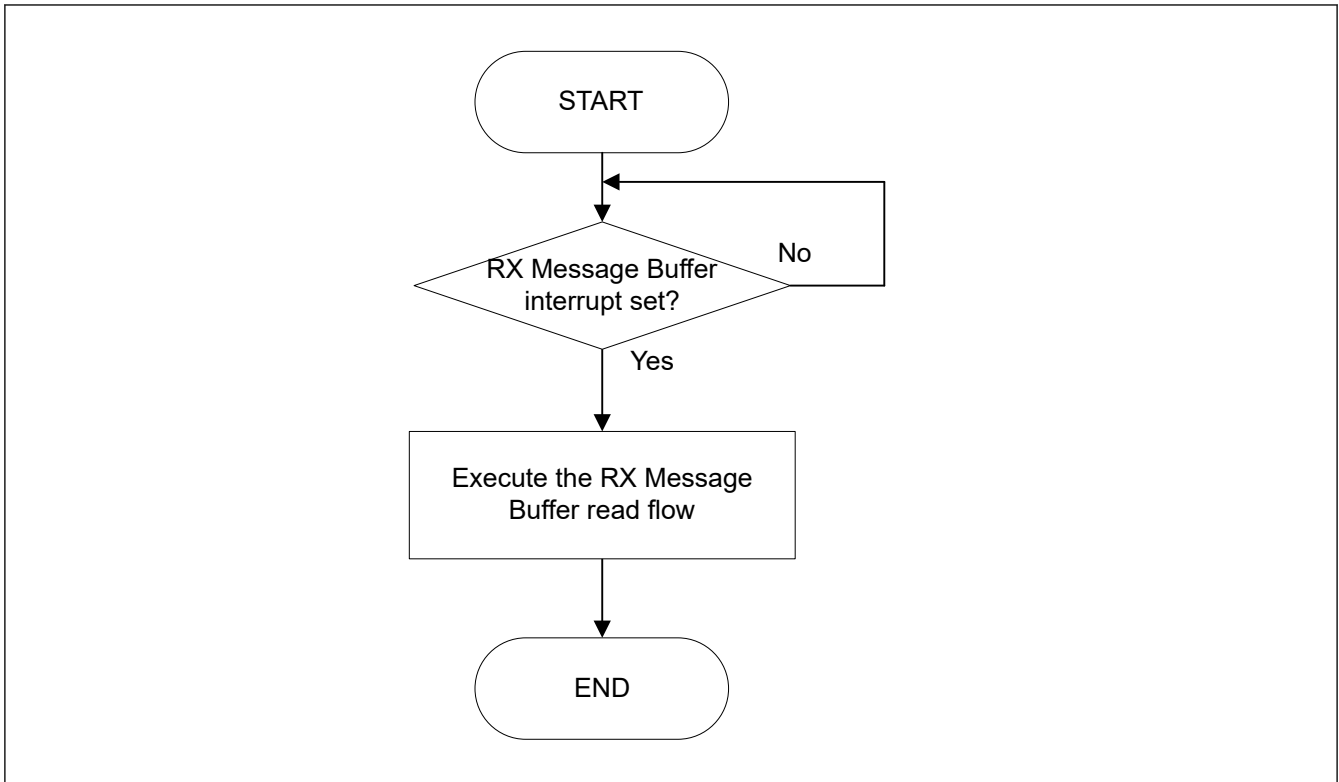
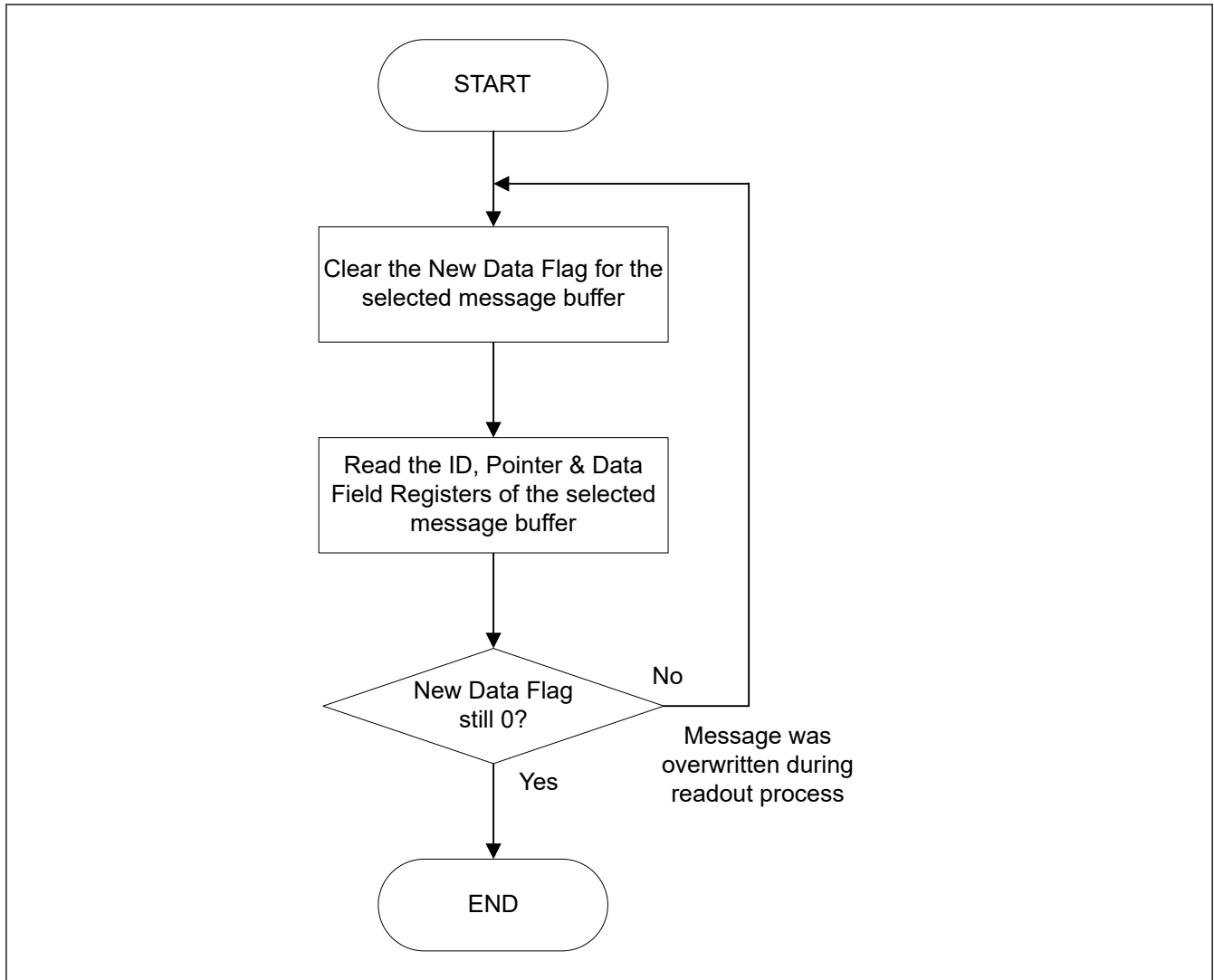


Figure 32.36 RX Message Buffer Message Access Flow (interrupt)



**Figure 32.37** Read flow of RX message buffer

### 32.8.1.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to RX FIFO buffers or Common FIFO buffer configured in RX mode should be configured based on system requirements.

The `CFDGAFLP1r.GAFLFDP[8,1:0]` field in the matching AFL entry selects the FIFO buffers to which the related reception message is stored.

When the received message is stored in one or more RX FIFO buffers or Common FIFO buffer configured in RX mode, the message counter value is incremented in the corresponding RX FIFO Status Registers or Common FIFO Status Register.

Depending on the configuration of the FIFO buffers, an interrupt might also be generated.

The message can be read from the corresponding FIFO Access registers.

**Note:** Because many messages can be stored in the FIFO buffers, reading more than one message may be required to read the latest message stored in a FIFO buffer.

If the message count value matches the FIFO depth, the FIFO Full flag is set.

When the value `0xFF` is written to the corresponding FIFO Pointer Control Register, the message count is decremented by 1.

Only write `0xFF` to the FIFO Pointer Control register after reading the complete message from the FIFO Access registers of the corresponding FIFO.

When all the messages stored in the FIFO are read, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO full condition), the FIFO Message Lost flag is set and the new message is lost (no overwrite of already stored messages takes place).

An appropriate value can be configured as warning level to generate an interrupt before the FIFO full condition occurs to avoid loss of a message due to an overrun condition.

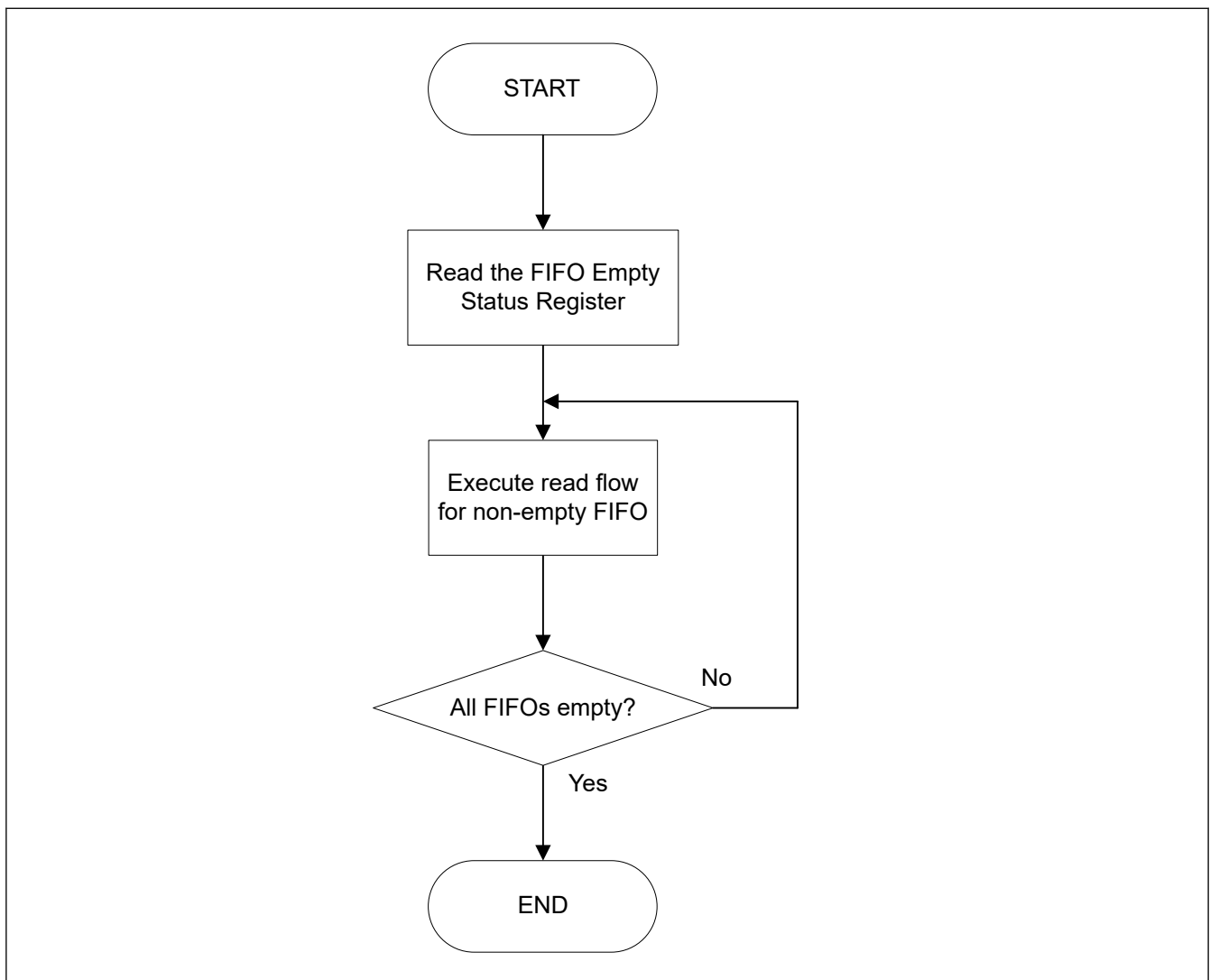
**Note:** The message lost can be set only in RX mode by CAN, and the flag is not set when the CPU is overloading the FIFO buffers.

The RX FIFO buffers and the Common FIFO buffers configured in RX mode can be disabled at any time by clearing the CFDRFCCa.RFE or CFDCFCC.CFE bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

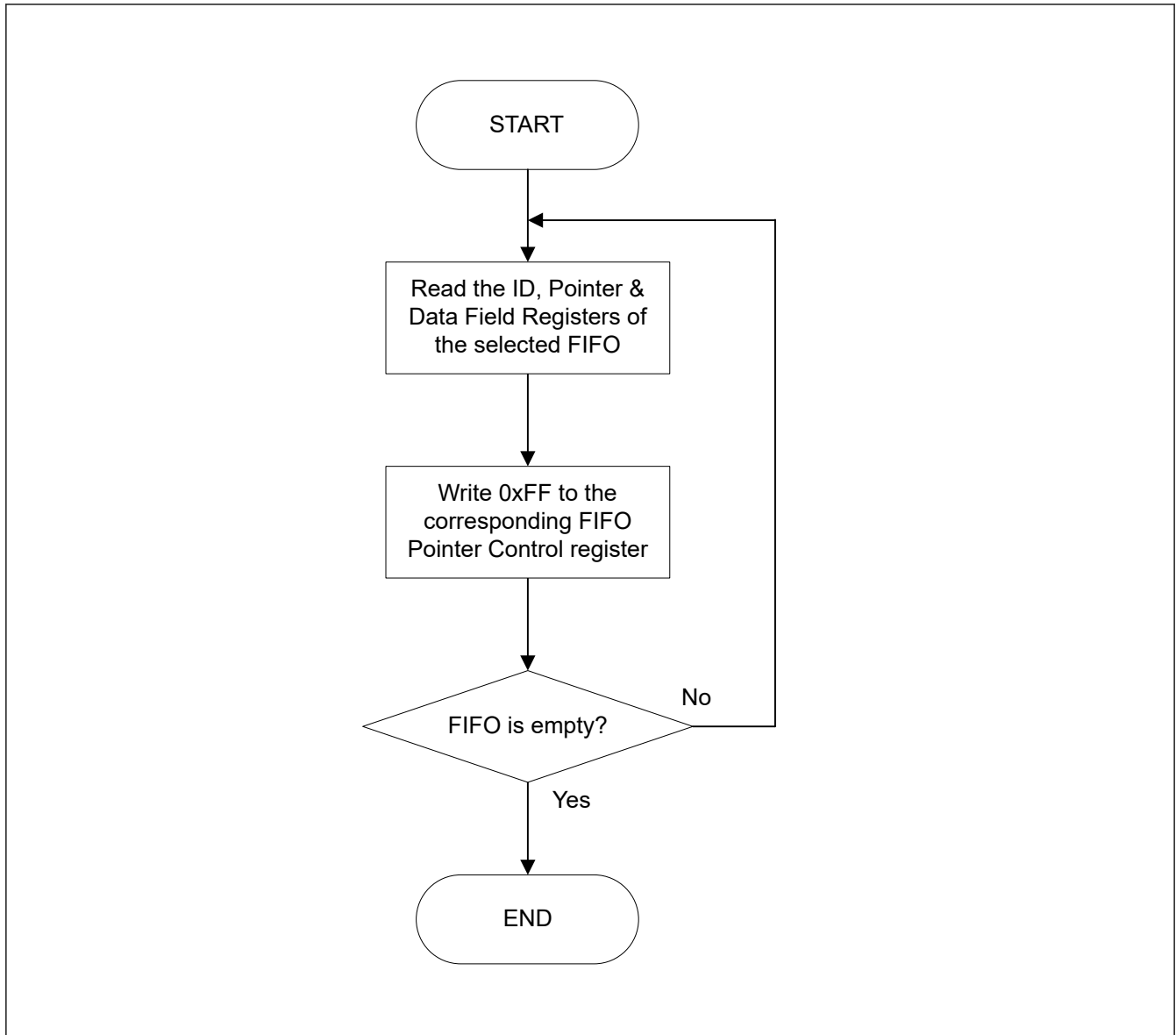
When the CFDRFCCa.RFE or CFDCFCC.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further messages can be stored into the FIFO.

When the RX FIFO buffers or Common FIFO buffer configured in RX mode is assigned as a DMA channel, software should not access the FIFO Access Register of this FIFO buffer or write 0xFF to the FIFO Pointer Control Register (CFDCFPCTR.CFPC or CFDRFPCTRa.RFPC). This can lead to unintended FIFO message decrement. The DMA channel controls the FIFO decrement automatically.

**Note:** If the interrupt flag is set for a FIFO buffer and then the FIFO is disabled, the interrupt flag is not cleared automatically. The interrupt flag should be cleared before disabling the FIFO.



**Figure 32.38** Access flow of FIFO buffer message (example for polling case)



**Figure 32.39 Read flow of RX FIFO buffer message (example for polling case)**

Note: When the next frame is received before clearing the completion interrupt flag of reception, the completion interrupt of reception is not set again.

Even when an interruption flag is cleared after the completion processing of reception, the already received interrupt flag is not set.

It is necessary to perform the completion processing of reception even before the next completion of frame reception, and to clear an interruption flag.

When processing does not meet the condition, after checking that receiving data is empty, interrupt flag is cleared and it checks that receiving data is empty again.

### 32.8.1.3 Timestamp

The timestamp counter is a free-running counter that can be used to check reception time of an incoming message or transmission time of successful transmitted messages. The Timestamp counter value is captured based on the `CFDGFDCFG.TSCCFG[1:0]` configuration (at the sample point of start of frame, point in time when the frame is valid, or for CANFD frames also at the sample point of the RES bit). For reception, it is stored together with the message ID and data into the target RX message buffer or RX FIFO.

For transmit message, the timestamp counter value is stored as part of the TX History List entry.

The counter can be clocked with the peripheral clock or with the CAN channel bit timing clock. The counter source clock can be configured with the CFDGCFG.TSSS bit of the Global Configuration Register. If this bit is 0, the peripheral clock is used. If the bit is 1, the selected CAN channel bit time clock is used.

The channel selection is performed with the CFDGCFG.TSBTCS bit of the Global Configuration Register.

Care must be taken when using selected CAN channel bit time clock as the clock source. When entering Channel Halt mode or Channel Reset node, for this channel, the timestamp counter is stopped. For other CAN channels, the timestamp counter value is not updated.

If peripheral clock is selected as the timestamp counter clock source, Channel modes do not affect the timestamp counter function.

The source clock for the timestamp counter can be divided by a factor defined by the CFDGCFG.TSP bits (timestamp prescaler) in the Global Configuration Register.

The timestamp counter can be reset to 0x0000 with the CFDGCTR.TSRST bit (timestamp reset).

### 32.8.2 Transmission

There are several possible transmission configurations:

- Normal transmission
- FIFO transmission
- TX Queue transmission

A fixed number of transmission message buffers (4 TX message buffers) are dedicated. These message buffers are only used for transmission and cannot be configured for reception.

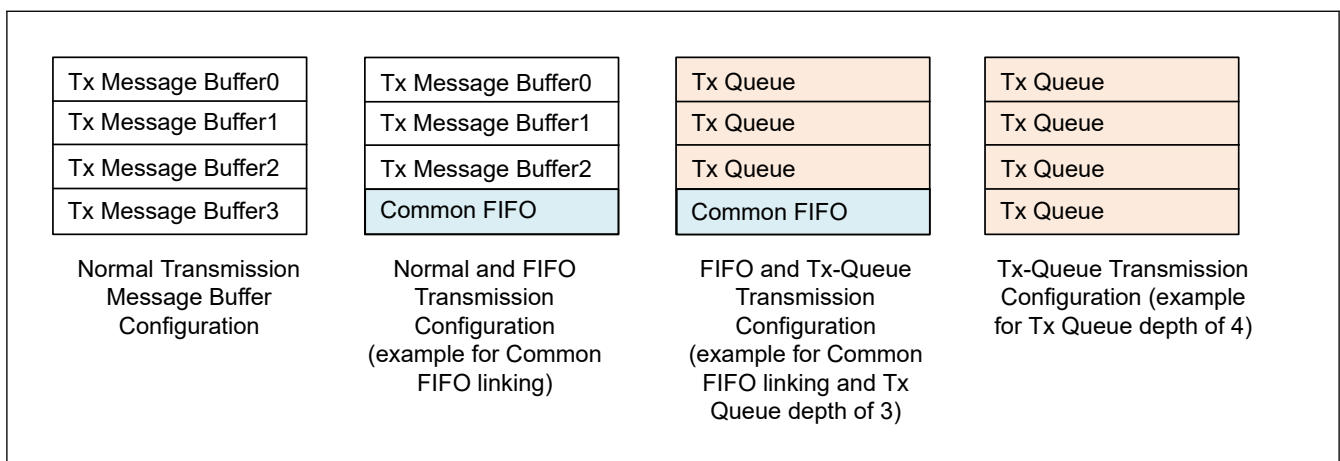
Additionally transmission from TX Queue or Common FIFO in TX mode can be configured in the following way (see [Figure 32.40](#)):

- TX Queue: Up to four transmission message buffers can be grouped to form a TX Queue with a common access window.  
Upper transmission message buffers are used to form the TXQ.  
TXQ has an access window.
  - TXQ is transmission Message Buffer 0.

- Common FIFO (TX mode): Common FIFO in TX mode is linked to a dedicated channel. Channel has a fixed number of one Common FIFO assigned to it. Within the channel, a Common FIFO configured in TX mode, can be freely linked (assigned) between 0 and 3 transmission message buffers (only one FIFO to one transmission message buffer).  
The Common FIFO buffer then replaces the transmission message buffer linked to it.  
Transmission Control and Status registers of these transmission message buffers should not be used.

See [Figure 32.28](#) for information about Common FIFO buffer assignment to related channel.

Note: Common FIFO buffer should not be linked to TX message buffers that are already part of a TX Queue.



**Figure 32.40 Configuration of channel transmission message buffer**

### 32.8.2.1 Transmission Priority

If two or more transmission message buffers of a channel are configured for transmission, then the transmission priority in the CANFD module can be selected from the following two modes:

- CAN ID priority
- Message buffer number priority.

The transmission priority mode is common for all message buffers. It can be configured with the `CFDGCFG.TPRI` bit in the Global Configuration Register.

For message buffer number priority transmission, the smallest message buffer number with transmission request has the highest priority for transmission. This also includes the TX message buffers linked to the Common FIFO buffer configured in TX mode.

However, message buffer number priority should not be used if TX Queue is enabled.

For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All TX message buffers can enter the ID priority comparison for message buffers configured for transmission. This also includes the TX message buffers linked to the Common FIFO buffer configured in TX mode and includes the TX Queue message buffers.

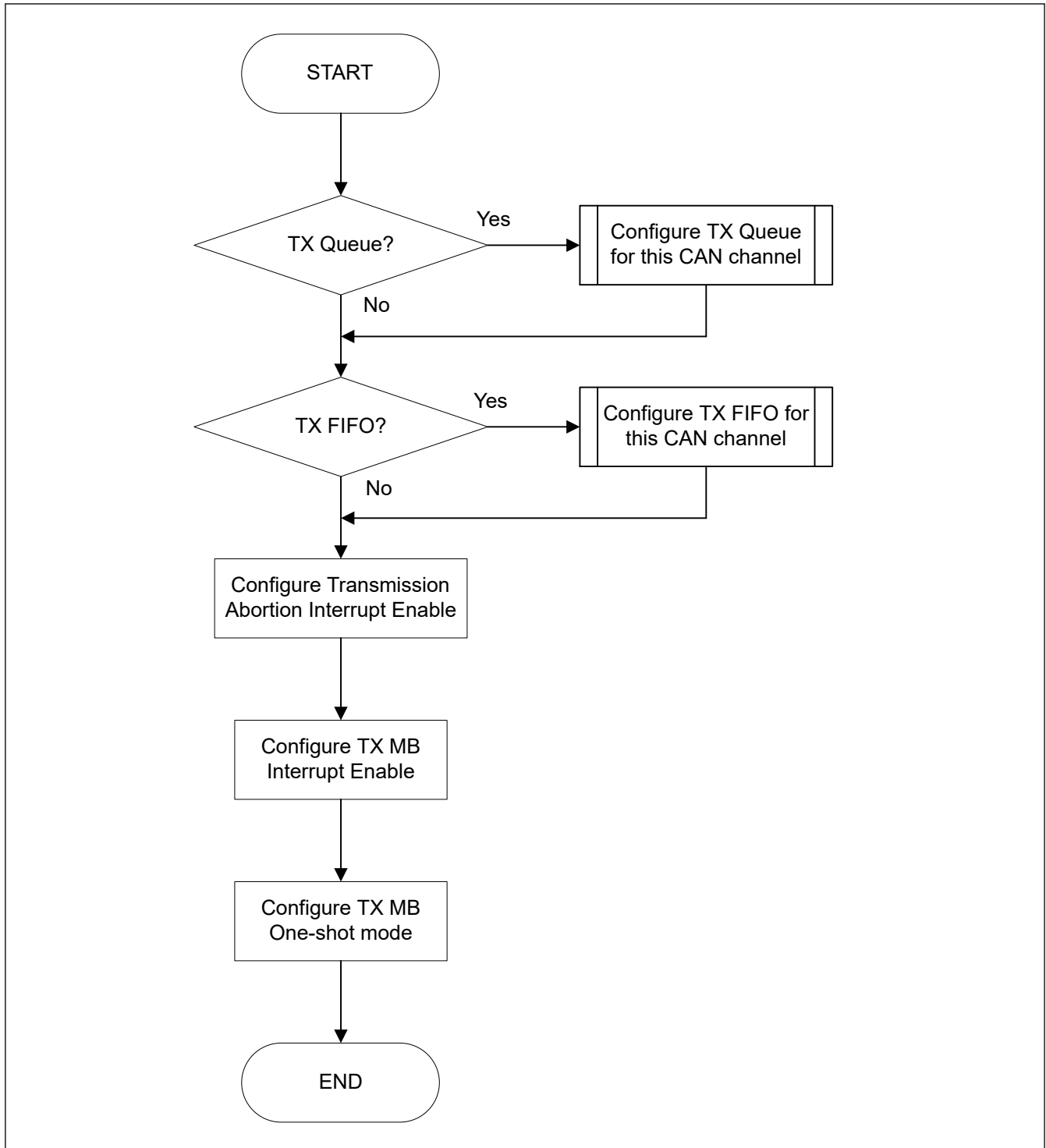
If the ID of two or more message buffers is the same, then the smaller message buffer number has higher priority for transmission.

**Note:** For Common FIFO buffer configured in TX mode, only the message currently being pointed to by the FIFO read pointer can be included in the transmission arbitration.

If the message is being transmitted from the FIFO, then the next pending message within the same FIFO is considered in the transmission arbitration.

In contrast to this, all transmission message buffers of a TX Queue participate in internal transmission arbitration.

[Figure 32.41](#) shows the transmission configuration flow.



**Figure 32.41** Flow for transmission configuration

### 32.8.2.2 Normal Transmission

Each transmission message buffer has two modes of message transmission:

1. Regular transmission mode

If the message buffer is placed in regular transmission mode, the data frame or remote frame set in that message buffer can be transmitted.

Completion of regular transmission can be checked through the related TX Message Buffer Transmission Result flag bits (CFDTMSTSj.TMTRF) in the TX Message Buffer Status Registers. These bits are set to 10b or 11b when the regular transmission is successful.

When arbitration is lost or an error occurs, message transmission is further attempted if no transmission abort request is set for this transmission message buffer.

New internal transmission arbitration for this channel is performed for all message buffers with transmission request.

2. One-shot transmission mode

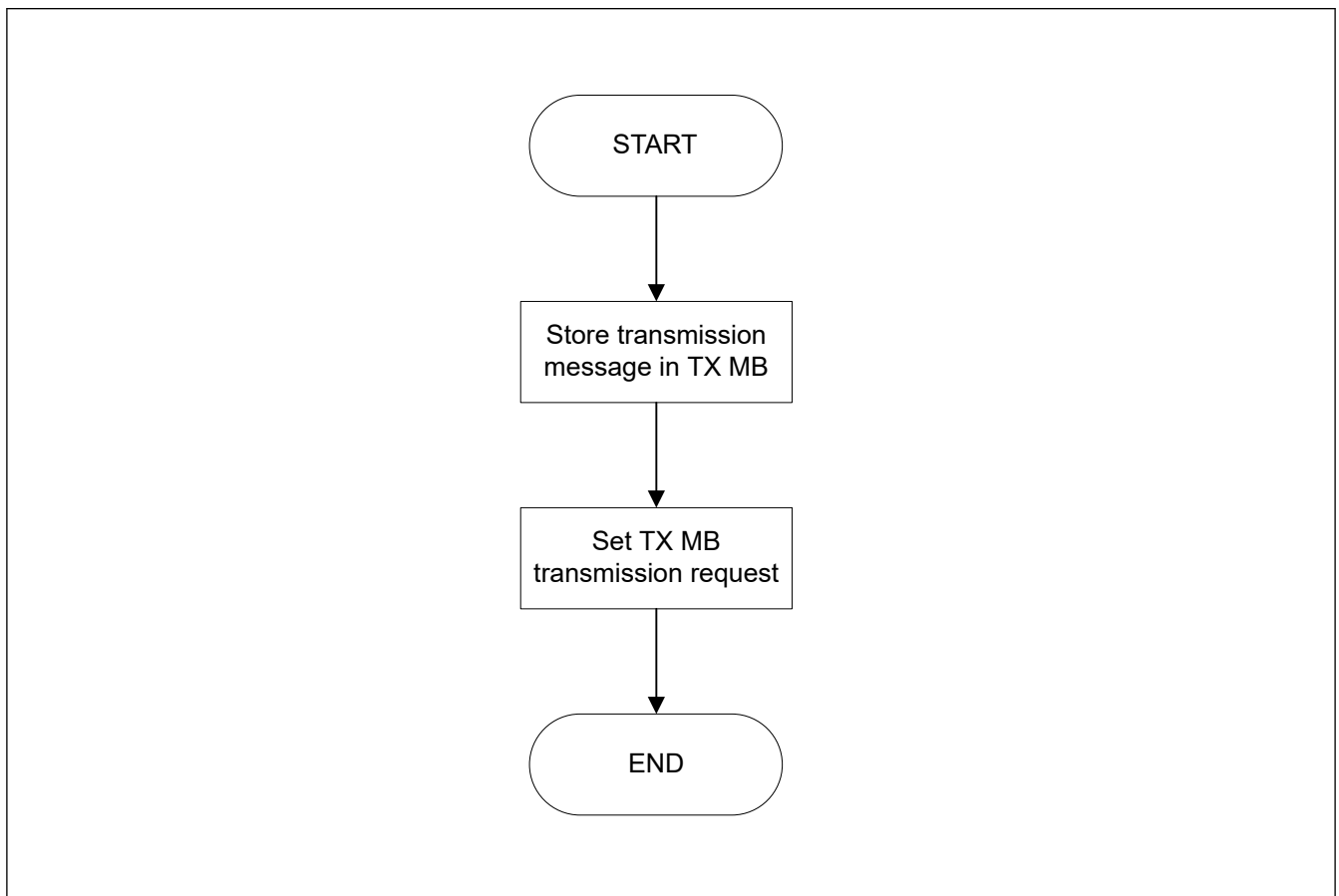
When the CFDTMCI.TMOM bit of the TX Message Buffer Control Registers is set for a transmission message buffer, the message buffer is placed in One-shot transmission mode and attempts to transmit a message only once.

Completion of One shot transmission can be checked through the related TX Message Buffer Transmission Result Flag bits (CFDTMSTSj.TMTRF) in the TX Message Buffer Status Registers. The CFDTMSTSj.TMTRF bits are set to 10b or 11b when One-shot transmission is successful.

The CFDTMSTSj.TMTRF bits are set to 01b when arbitration is lost or an error occurs during transmission of the related message buffer.

Additional message transmission is not attempted in this case.

The regular transmission request procedure after a configuration is shown in [Figure 32.42](#).



**Figure 32.42** Transmission request procedure using normal TX Message Buffer mode

(1) Setting for TX Message Buffer Control Register

[Table 32.26](#) shows configuration of a normal CAN transmission mode.

**Table 32.26** Configuration of CAN transmission mode (1 of 2)

Transmission request CFDTMCI.TMTR	Transmission abort request CFDTMCI.TMTAR	One-shot enable CFDTMCI.TMOM	Communication activity
0	0	0	Message buffer disabled
0	0	1	Message buffer disabled

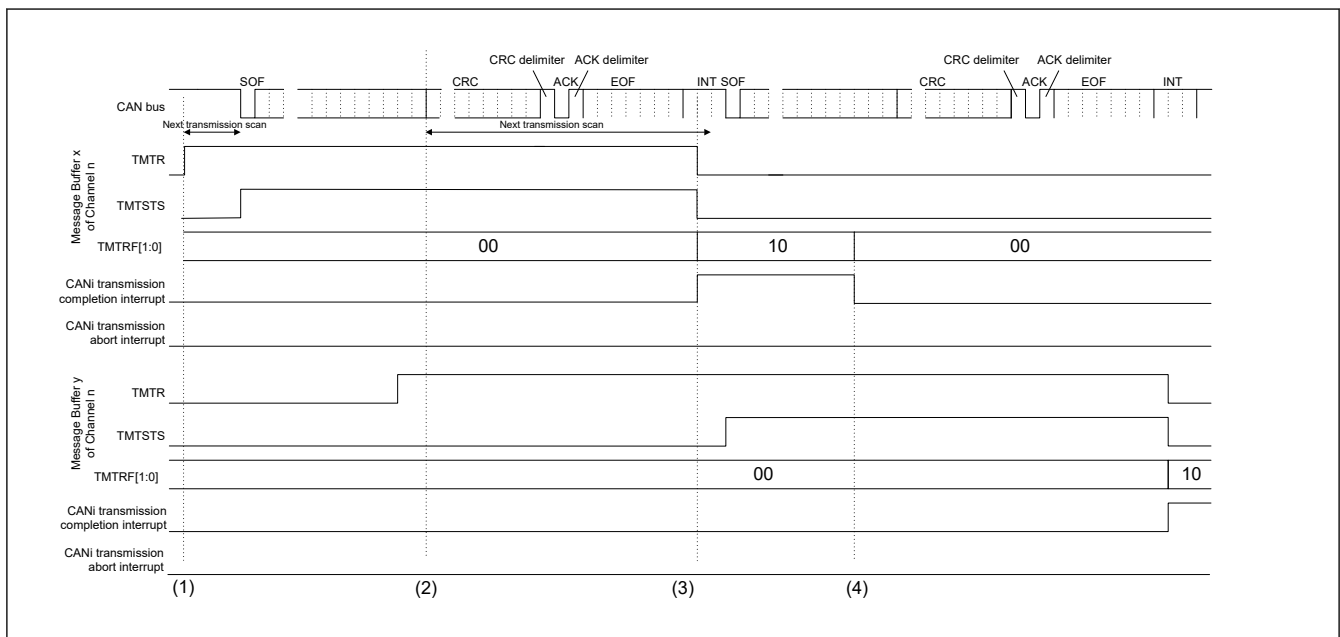


**Table 32.26 Configuration of CAN transmission mode (2 of 2)**

Transmission request CFDTMCi.TMTR	Transmission abort request CFDTMCi.TMTAR	One-shot enable CFDTMCi.TMOM	Communication activity
1	0	0	Configured as a transmission message buffer for a data frame or a remote frame
1	0	1	Configured as a one-shot transmission message buffer for a data frame or a remote frame
1	1	0	Transmission abort requested
1	1	1	One-shot transmission abort requested

The configuration bits can be configured in the TX Message Buffer Control Registers.

Figure 32.43 shows timings for successful transmission for two message buffers.



**Figure 32.43 Timing of request and flag bits for successful transmission**

1. If the CFDTMCi.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, the message buffer scanning procedure determines the highest priority message buffer for transmission. When the transmission message buffer is determined, the CFDTMSTSj.TMTSTS bit in the related TX Message Buffer Status Registers is set (transmitting/transmitter), and CAN channel starts the transmission <sup>\*1</sup>.
2. At the first bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.
3. If the message has been successfully transmitted, the CFDTMSTSj.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 10b and CFDTMSTSj.TMTSTS and the CFDTMCi.TMTR bits are cleared. When the TMIE bit in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated. To clear the related interrupt line, clear the CFDTMSTSj.TMTRF flag bits.
4. Before starting the next transmission, clear the CFDTMSTSj.TMTRF bits. Load the next message in the transmission message buffer and set the CFDTMCi.TMTR bit again. CFDTMCi.TMTR bit cannot be set again before CFDTMSTSj.TMTRF[1:0] bits are cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSj.TMTSTS bit is cleared. The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit. If an error occurs either during transmission or following the loss of arbitration, then during the error frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

Note: The setting point of CFDTMSTSj.TMTSTS is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID due to the synchronization logic implemented for the PLL bypass.

Figure 32.44 shows timings for transmission abort for two message buffers.

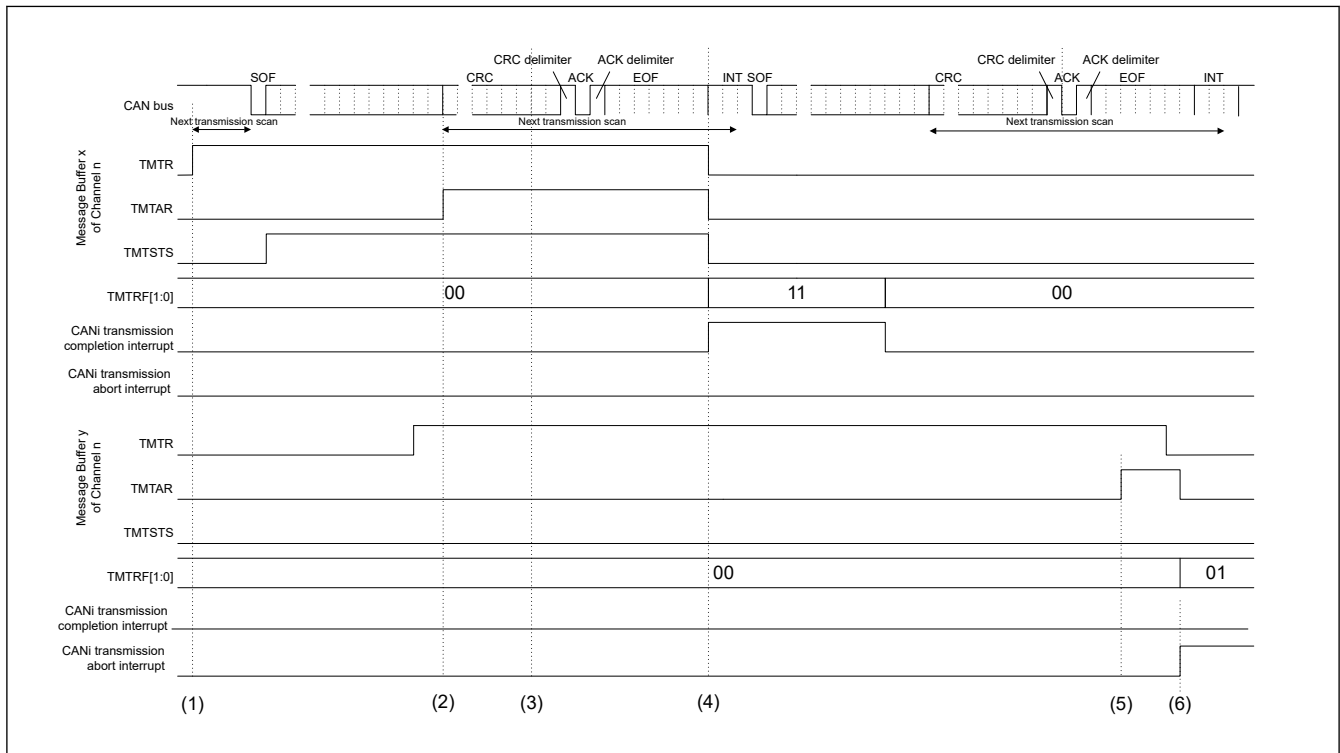


Figure 32.44 Timing of request and flag bits for transmission abort

1. If the CFDTM<sub>Ci</sub>.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, the message buffer scanning procedure determines the highest priority message buffer for transmission. When the transmission message buffer is determined, the CFDTMSTS<sub>j</sub>.TMTSTS bit in the TX Message Buffer Status Registers is set (transmitting/transmitter), and CAN channel starts the transmission\*1.
2. If the CFDTM<sub>Ci</sub>.TMTAR bit is set when the related message buffer is already selected for transmission or currently transmitting, the message is not aborted, if no error occurs or arbitration is lost.
3. At the first CRC bit, the transmission scanning procedure starts for the next transmission. In this example, timing chart message buffer y is not selected as the next transmission message buffer.
4. If the message has been successfully transmitted, the CFDTMSTS<sub>j</sub>.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 11b and the CFDTMSTS<sub>j</sub>.TMTSTS and CFDTM<sub>Ci</sub>.TMTR bits are cleared. When the TMIE bit in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated. To clear the related interrupt line, clear the CFDTMSTS<sub>j</sub>.TMTRF[1:0] bits.
5. Another CAN node is transmitting on the CAN bus (CFDTMSTS<sub>j</sub>.TMTSTS is not set). If the CFDTM<sub>Ci</sub>.TMTAR bit is set when the related channel is under transmission scan, the transmission request cannot be cleared.
6. After internal processing time, the transmission is aborted and the CFDTMSTS<sub>j</sub>.TMTRF[1:0] bits are set to 01b. If the message buffer is not transmitting or selected as the next transmission message buffer or under transmit scan, then the abort is immediately accepted and the corresponding CFDTMSTS<sub>j</sub>.TMTRF[1:0] bits in the TX Message Buffer Status Registers are set to 01b. In addition, CFDTM<sub>Ci</sub>.TMTR, and CFDTM<sub>Ci</sub>.TMTAR bits are cleared automatically. When the transmission abort interrupt enable TAIE bit of the related Channel Control Register is set then an interrupt is generated for successful transmission abort. To clear the related interrupt line the CFDTMSTS<sub>j</sub>.TMTRF[1:0] bits have to be cleared.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTS<sub>j</sub>.TMTSTS bit is cleared.

The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit.

If an error occurs, either during transmission, or following the loss of arbitration, then during the error frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

### 32.8.2.3 TX FIFO Transmission

One common FIFO buffer is assigned to CANFD module. The FIFO buffer can be linked to any normal TX message buffer position for this channel with the CFDCFCC.CFTML bits in the Common FIFO Configuration/Control Register if configured in TX mode.

When the transmission scan starts and the FIFO buffer corresponding to this TX message buffer is enabled, the relevant message in the FIFO buffer participates in the transmission scan.

Configuration of a TX message buffer linked to a FIFO buffer configured in TX mode should not be done.

#### (1) TX FIFO Operation

CAN messages can be written into the TX FIFO by writing to the corresponding FIFO Access registers.

When the value 0xFF is written into the corresponding FIFO Pointer Control Register, the message count of the related FIFO is incremented by 1.

Only write to the FIFO Pointer Control register after writing the complete message to the corresponding FIFO Access registers. If the message count matches the FIFO depth, the FIFO Full flag is set.

The oldest message in the TX FIFO is included in the scan for transmission by the corresponding CANFD module channel logic.

When a message is successfully transmitted from the TX FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the FIFO Empty flag is set.

The interrupt generation conditions for the TX FIFO buffer can be configured by configuring the CFDCFCC.CFIM bit in the corresponding Common FIFO Configuration/Control Register.

If CFDCFCC.CFIM bit is 0, then interrupt is generated when the last message is successfully transmitted from the TX FIFO buffer.

If CFDCFCC.CFIM bit is 1, then interrupt is generated for every successfully transmitted message from the TX FIFO buffer.

The Common FIFO can set interrupt when CAN frame transmission is complete.

The Common FIFO buffer configured in TX Mode can be disabled by clearing the CFDCFCC.CFE bit in the Common FIFO Configuration/Control Register. If this bit is cleared to 0, the FIFO Empty flag is set as follows:

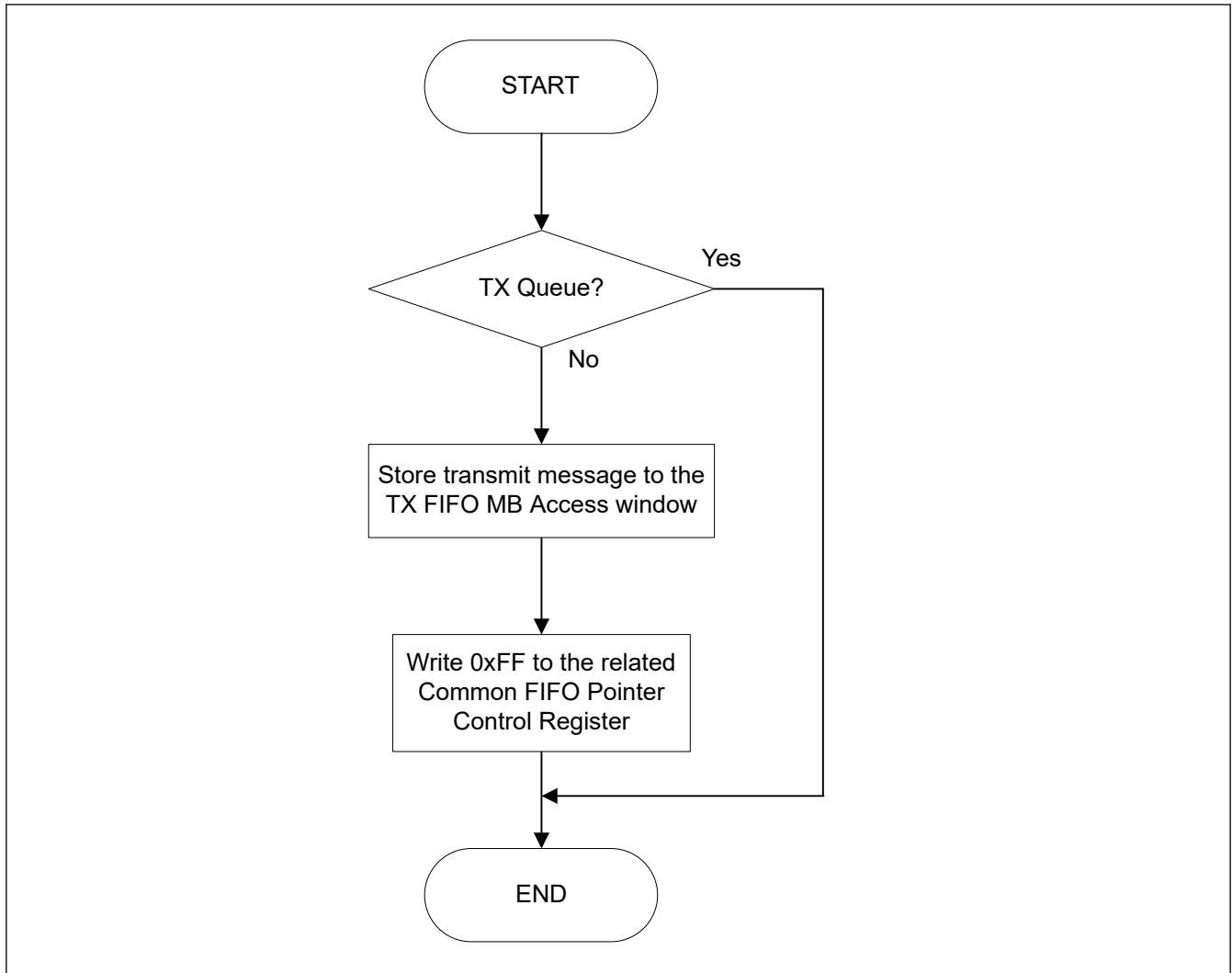
- Immediately if the message from the TX FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX FIFO is already scheduled for transmission or already in transmission.

**Note:** The Common FIFO buffer is considered as disabled after clearing the CFDCFCC.CFE bit only when the Empty flag is set for the corresponding Common FIFO buffer.

Other possible messages pending from the TX FIFO are lost and their transmission must be requested again. Before CFDCFCC.CFE is set again, ensure that CFDCFSTS.CFEMP bit is set and that there are no pending abort from the TX FIFO.

When the CFDCFCC.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after configuration is shown in [Figure 32.45](#).



**Figure 32.45 Request procedure for TX FIFO transmission**

## (2) Interval Timer for FIFO Transmission

For each Common FIFO in TX mode, it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the CFDCFCC.CFE bit is set.

When the Common FIFO in TX mode is enabled, the first message is transmitted without considering this interval time.

The interval timer stops counting when:

- FIFO is disabled by clearing the CFDCFCC.CFE bit.
- CAN channel is in CH\_RESET mode.

The interval time is specified by the CFDCFCC.CFITT value from 0 to 255 timer units in the Common FIFO Configuration/Control Register.

The timer unit can be defined based on two different source clocks for the interval timer. To disable the interval timer for FIFO transmission, select a value of 0.

The timer source can be selected with the configuration bit CFITSS in the Common FIFO Configuration/Control Register.

If CAN channel bit time clock is configured as the clock source, and the CAN channel enters CH\_HALT, CH\_RESET, or CH\_SLEEP mode, the interval timer is stopped for that channel.

If peripheral clock is selected as the interval timer clock source, the interval timer is stopped only when the CAN channel is in CH\_RESET or CH\_SLEEP mode.

The reference clock can be used to configure the interval time in fixed time units. It is based on the peripheral clock. The reference clock prescaler value `CFDGCFG.ITRCP` in the Global Configuration Register defines the relation between the peripheral clock frequency/period and the reference clock period.

See [Table 32.27](#) for `CFDGCFG.ITRCP` configuration values to achieve different reference clock periods based on the peripheral clock frequency and period.

**Table 32.27 Configuration example for the reference clock of the FIFO interval timer**

Reference clock/Peripheral clock	1 $\mu$ s	100 $\mu$ s	500 $\mu$ s
16 MHz/62.5 ns	16	1600	8000
20 MHz/50 ns	20	2000	10000
32 MHz/31.25 ns	32	3200	16000
50 MHz/20 ns	50	5000	25000

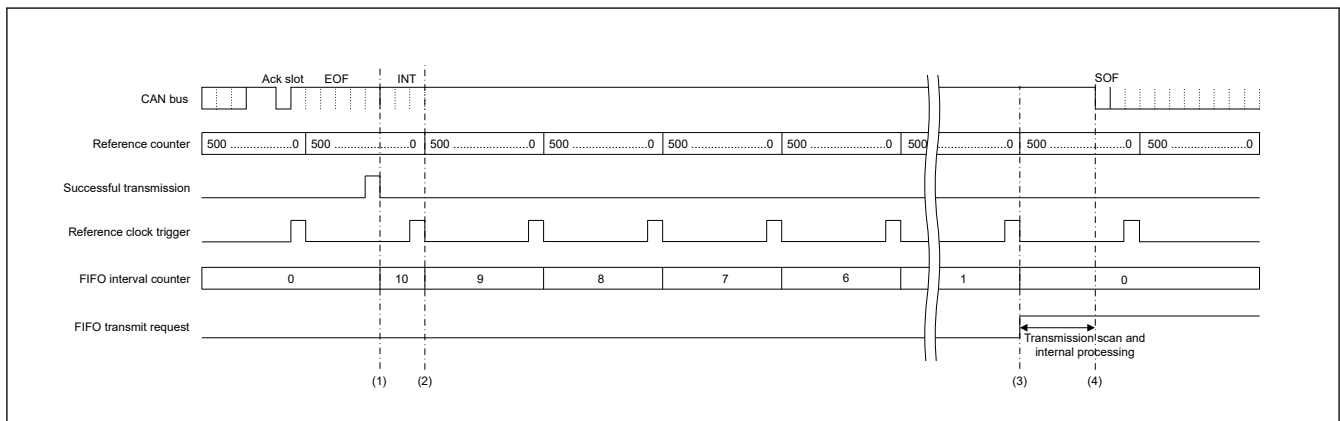
The reference clock resolution can be specified by the interval timer reference clock resolution value `CFDCFCC.CFITR` in the Common FIFO Configuration/Control Register.

The interval time is based on the reference clock period multiplied by the configured value (x1 or x10). The reference clock based interval timer can be used to satisfy the requirements of the ISO 15765-2 Separation Time. The whole range for the separation time from 100  $\mu$ s to 127 ms can be covered.

The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).

When the interval time has elapsed, the next transmission request is raised by the related TX FIFO. Therefore, the interval time defines the minimum time between two messages transmitted from one FIFO.

The next message is sent at earliest after this interval time. [Figure 32.46](#) shows an example timing of the internal processing.



**Figure 32.46 Example for interval processing time**

The configuration for the timing in [Figure 32.46](#) is as follows:

- Peripheral clock frequency = 50 MHz
- Interval timer reference clock (`CFDGCFG.ITRCP`) = 500 times
- Reference clock from the settings in [Figure 32.46](#) = 10  $\mu$ s
- Common FIFO interval timer source selection (`CFDCFCC.CFITSS`) = 0
- Common FIFO interval timer resolution (`CFDCFCC.CFITR`) = 0
- Common FIFO interval transmission time (`CFDCFCC.CFITT`) = 10 times
- Theoretical message separation interval = 100  $\mu$ s

1. Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the reference clock trigger. Therefore, the first interval is counting less or equal to 1 reference clock interval.
2. With the next reference clock trigger the FIFO interval timer is decremented.

3. When the FIFO interval timer reached the value 0, the FIFO transmit request is set.
4. When the FIFO is selected for transmission, the transmission starts. Due to internal processing, this usually takes less than 3 CAN bit time, between the internal FIFO transmit request set in step 3. and the actual transmission.

In the worst case when multiple events such as a reception scan, an internal message routing, a transmit scan on all channels occur, it can take up to 120 peripheral clock cycles.

As shown in Figure 32.46, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, configure CFDCFCC.CFITT to the required minimum value plus 1.

If additional TX message buffers or TX FIFO are configured for transmission of the same channel, the real delay between two messages transmitted from a TX FIFO can be much longer than specified by the interval time. This is due to higher priority message transmission from these TX message buffers or TX FIFO.

Figure 32.47 shows a block diagram of the FIFO interval time generation circuit.

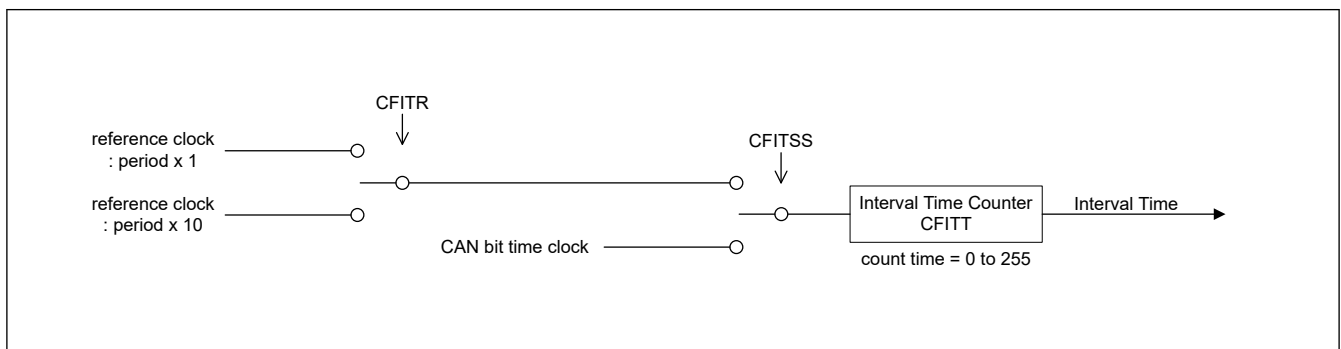


Figure 32.47 Block diagram of FIFO interval timer

### 32.8.2.4 TX Queue

Each enabled TX Queue for a specific channel consists of 3 to 4 TX message buffers, which are accessed through one access window.

- The first TX Queue can be configured with a depth of three up to four buffers and uses TX Message Buffer No. 0 as access window (referred to as TXQ)

All the TXQ messages enter the priority comparison for the transmission, which should be only ID Priority (CFDGCFCFG.TPRI = 0).

The registers for TXQ are:

- CFDTXQCC
- CFDTXQSTS
- CFDTXQPCTR

See related access registers TX Message Buffer ID Registers (TMID[m]), TX Message Buffer Pointer Registers (TMPTR[m]), TX Message Buffer Data Field 0 Registers, and TX Message Buffer Data Field 1 Registers (TMDF[0:1][m]) when access window TXQ0 is used.

The depth of each TXQ buffer can be configured by writing to the CFDTXQCC.TXQDC[1:0] bits of the TX Queue Configuration/Control Register. TXQ can be set from TXMB0 to TXMB3 as a queue buffer at the maximum.

The 4 available options for the depth configuration of TXQ buffer are:

- 0x00: TX Queue disabled
- 0x01: reserved
- 0x10: 3 Messages
- 0x11: 4 Messages

Do not access all the TX message buffers forming the TX Queue directly (except TX Message Buffer No. 0, which act as TX Queue access window).

When a system writes in TXQ, it writes in send data, after checking the state of TXQ.

Do not access or configure the related TX Message Buffer Control Registers.

The messages stored to the TX Queue access window are internally stored to a free buffer of the TX Queue.

When the buffer is full, no further access should be done to the queue, until it is no longer full. If access is a software write when the buffer of TXQ is full, send data is overwritten.

The TX Queue can be disabled by clearing the TXQE bit in the TX Queue Configuration/Control Register. If this bit is cleared, the TX Queue Empty flag is set as follows:

- Immediately if the message from the TX Queue is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX Queue is already scheduled for transmission or already in transmission.

Note: The TX Queue is disabled only when the Empty flag is set after clearing the TXQE bit for the corresponding TX Queue.

Other possible messages pending from the TX Queue are lost and their transmission must be requested again.

Before TXQE is set again, ensure that the CFDTXQSTS.TXQEMP bit is set and that there is no pending abort from the TX Queue.

When the TXQE bit is cleared, all messages in the TX Queue buffers are lost and no further message should be stored in the TX Queue.

When a message has been stored to the TX Queue, write 0xFF in the TX Queue Pointer Control Register. This sets the transmit request automatically and changes the internal message buffer pointer to the next free message buffer location of the TX Queue.

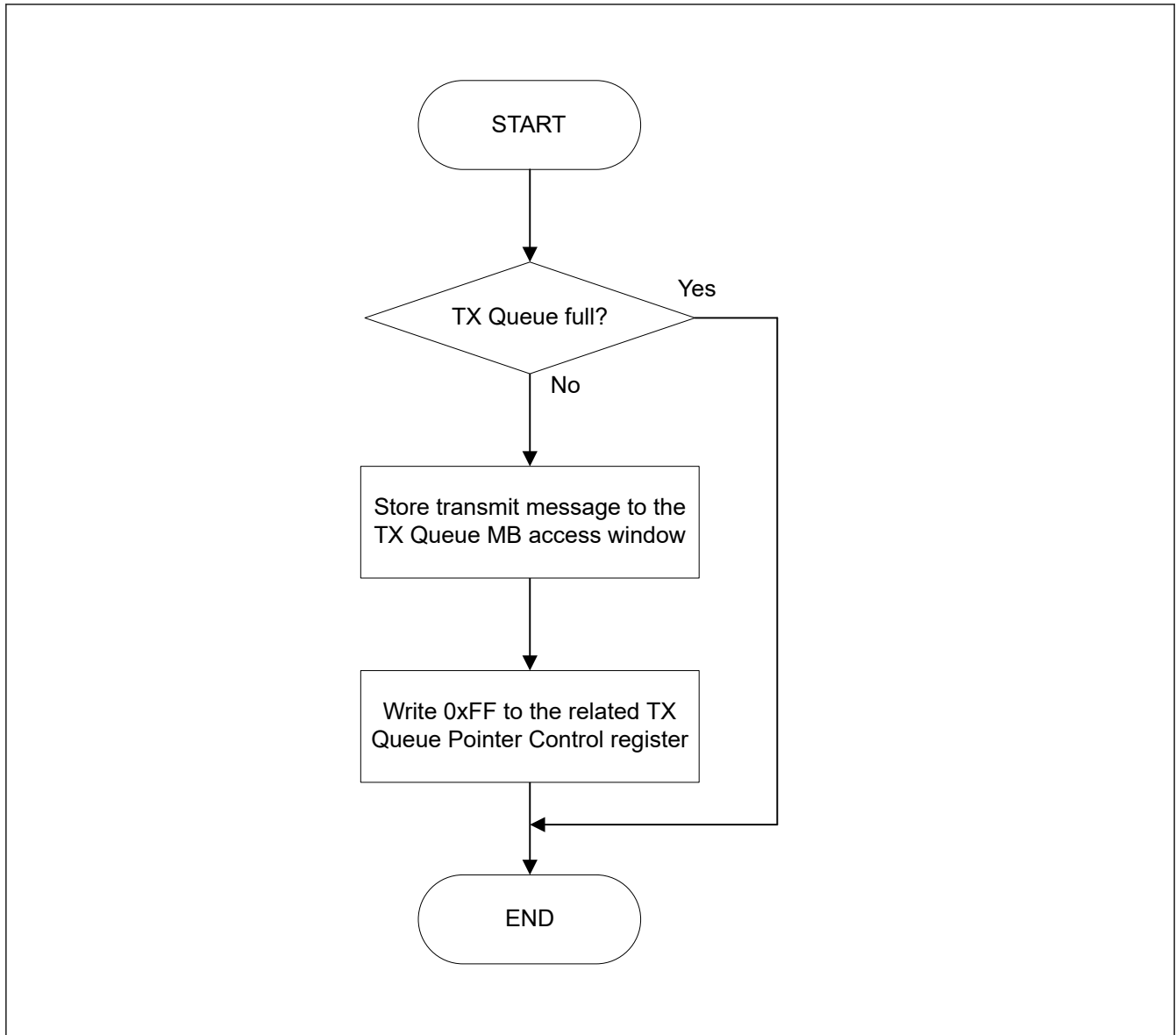
Note: If two messages with the same ID are stored in the TX Queue, the order of transmission of these messages can be different from the order in which they were stored in the TX Queue.

To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new message with the same ID is stored in the TX Queue.

For the TX Queue, a dedicated interrupt can be enabled by setting the TXQIE bit of the TX Queue Configuration/Control Register.

The interrupt mode can be configured with the CFDTXQCC.TXQIM bit of the same register either to generate an interrupt for every transmitted message or for the last transmitted message.

The TX Queue transmission request procedure after configuration is shown in [Figure 32.48](#).



**Figure 32.48 TX Queue transmission request**

### 32.8.2.5 TX History List

The TX History List function records the information of the successfully transmitted message in the TX History List Buffers. Two TX History List buffers are provided and THL buffer can store up to 8 TX History List entries.

The CFDTHLCC.THLDTE bit of the TX History List Configuration/Control Register can be used to configure if only message information from TX FIFO or TX Queue is stored, or if all transmit message information from TX Queue, TX FIFO, or normal TX message buffers is stored in the TX History List.

Each transmit message can be individually configured for acceptance to the TX History List with the CFDCFID.THLEN bit in the Message Buffer Pointer Register.

The message information is stored to the TX History List Buffer of a CAN channel after the message is successfully transmitted.

Storing to the list is not synchronized with the status of CFDTMSTSj.TMTRF[1:0] bits in the TX Message Buffer Status Register.

Due to internal processing, the storage to the list can happen with a delay after the successful transmission indication.

Storing the TX History List data can be recognized by the condition that the THLIF is set to 1 when the THLIE bit is configured to 1 or when the TX History List counter CFDTHLSTS.THLMC[5:0] is increased.



In worst case when multi events like reception scan, internal message routing on happen.

- Maximum delay time from setting the CFDTMSTSj.TMTRF to store the TX History List data is 70 peripheral bus clock cycles.

The History list records the following information of a transmitted message:

- Buffer type:
  - 001: TX Message Buffer
  - 010: TX FIFO
  - 100: TX Queue
- Buffer number:  
TX message buffer, TX Queue message buffer or TX message buffer link for the Common FIFO buffer from which transmission occurred. The number depends on the buffer type. See [Table 32.28](#).
- Transmission ID:  
Transmission pointer stored in the transmission message
- Transmit timestamp:  
Message timestamp captured at capture point as configured by CFDFDCFG.TSCCFG.
- Transmission information label:  
Transmission information label stored in the transmission message.

**Table 32.28 TX History List Buffer number entry**

Buffer Number	BT[2:0] Buffer Type		
	001b TX Message Buffer	101b TX FIFO	100b TX Queue
00b	Message Buffer 0	Number shown corresponds to the common FIFO. TX Message Buffer Link CFTML of the related Common FIFO configuration	Number shown corresponds to the Message Buffer belonging to the TX Queue which the frame was transmitted
01b	Message Buffer 1		
10b	Message Buffer 2		
11b	Message Buffer 3		

The Transmission ID entry is used to identify which message of a TX FIFO or TX Queue has been successfully transmitted because the TX FIFO or TX Queue number alone is not sufficient.

Therefore, a unique number can be attached to each transmission message stored in a TX FIFO or TX Queue. This unique identification number should be written to the CFDFDSTS.CFPTR[15:0] part of the Common FIFO Access Pointer Register for a TX FIFO or to the CFDTMFDCTRb.TMPTR[15:0] part of the TX Message Buffer Pointer Register of the TX Queue access window message buffer.

When the message is successfully transmitted, this identification number is stored together with the other message related information to the TX History List and can be read using the Transmission ID (TID) of the TX History List Access Register.

Also, for normal TX message buffers, the CFDTMFDCTRb.TMPTR[15:0] part of the TX Message Buffer Pointer Register is stored in the Transmission History List and the information label is the same.

[Figure 32.49](#) shows a transmission preparation flow when TX History List is used.

Read access to the TX History List Access Register is done for every single entry.

After reading one entry, 0xFF must be written to the corresponding TX History List Pointer Control Register to be able to access the next entry until TX History List is empty.

[Figure 32.50](#) shows an example flow for processing the TX History List information.

The TX History Lists have dedicated interrupts, which can be configured with the CFDTHLCC.THLIM bit of the corresponding TX History List Configuration/Control Register and enabled with the CFDTHLCC.THLIE bit of the same

registers, either to generate an interrupt when the History List reached a filling level of 75% or for every new TX History List entry.

An entry lost indication is flagged by the CFDTHLSTS.THLELT bit in the TX History List Status Register. The status of this bit is also shown by the THLES bit in the Global Error Flag Register.

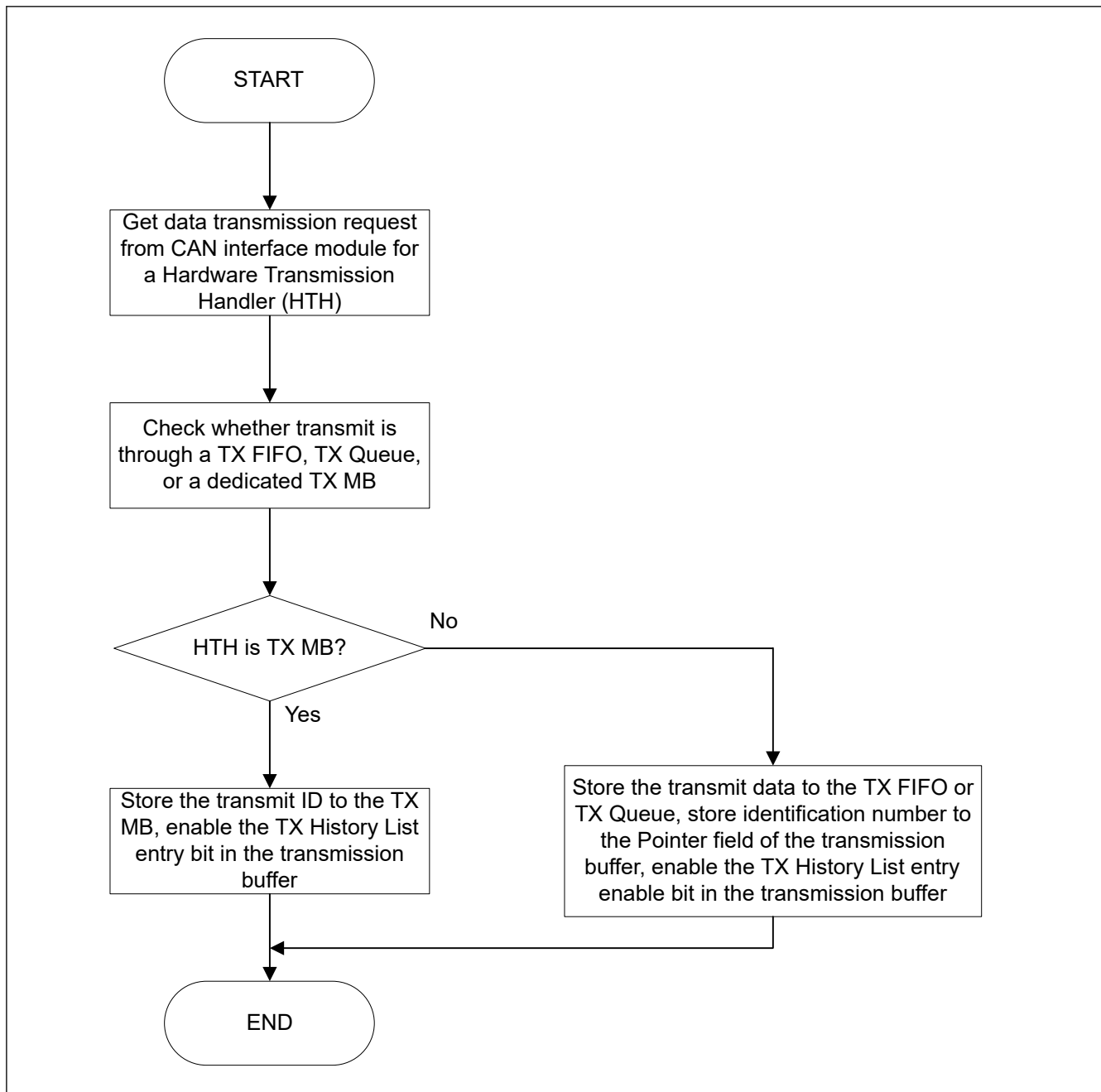


Figure 32.49 TX History List preparation flow

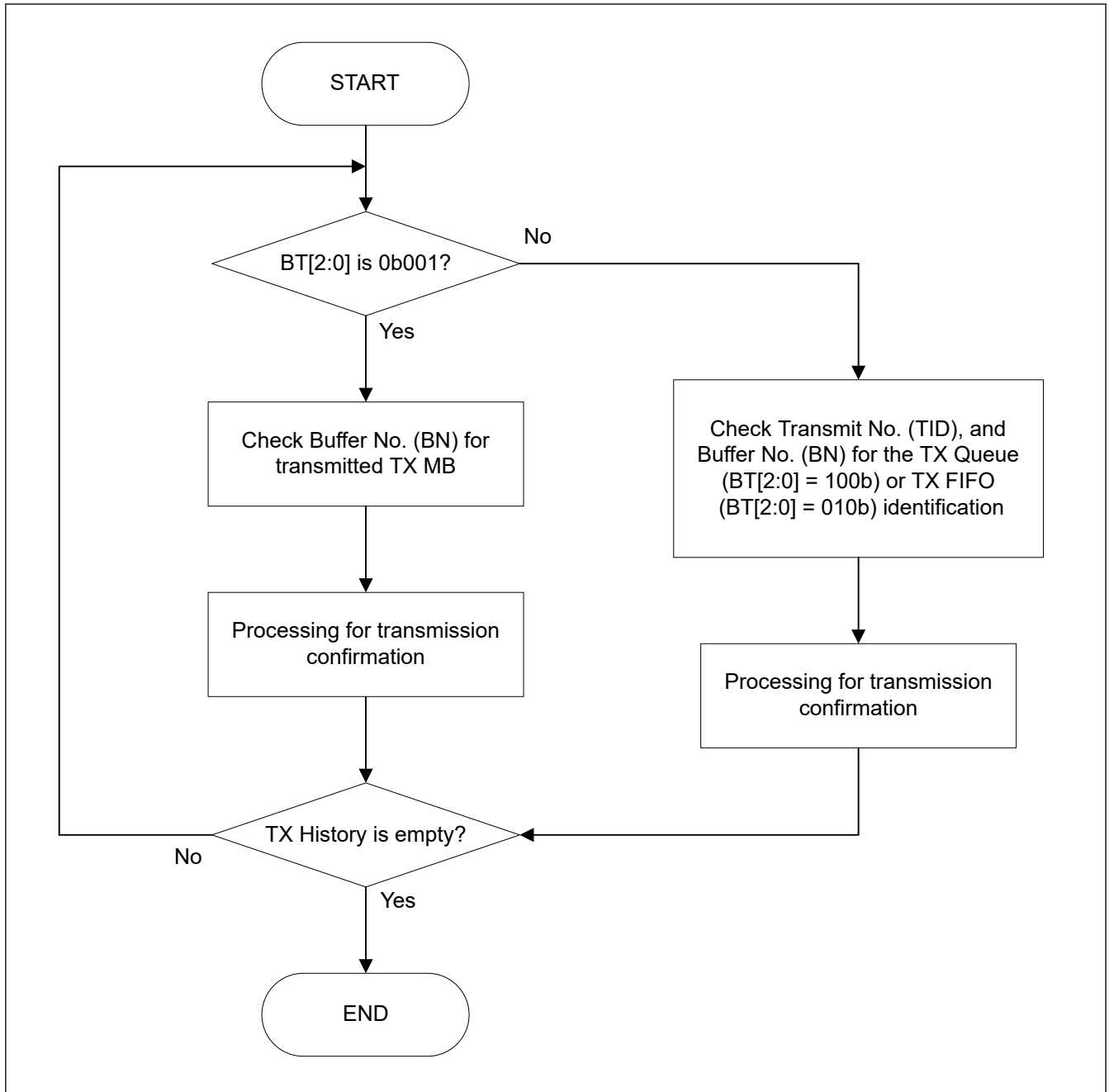


Figure 32.50 TX History List processing flow

### 32.8.2.6 TX Data Padding

This chapter is not valid for classical CAN.

If the data length code (DLC) of the transmitting message has a higher number of data bytes than the buffer size, the data bytes beyond the restricted range are replaced by bytes with the value of 0xCC.

This can happen for Common FIFO configured as (TX mode) when the transmit message DLC is higher than the CFDCFCC.CFPLS.

This can also happen in FD only mode, if a Classical frame is configured with a DLC bigger than 8.

## 32.9 Test Mode

The CANFD module can be configured into test modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the CANFD module in test modes.

Note: All test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable any combination of the various test modes specified in this section.

The test modes can be broadly split into 2 groups:

- Channel specific test modes
- Global test modes.

### 32.9.1 Channel Specific Test Modes

CAN channel can be configured into the following test modes:

- Basic test mode
- Listen-only mode
- Self-test mode 0 (External loop back mode)
- Self-test mode 1 (Internal loop back mode)
- Restricted operation mode.

#### 32.9.1.1 Basic Test Mode

The basic test mode should be used when there is requirement for a particular test setting to be enabled other than when in Listen-only and Self-test modes.

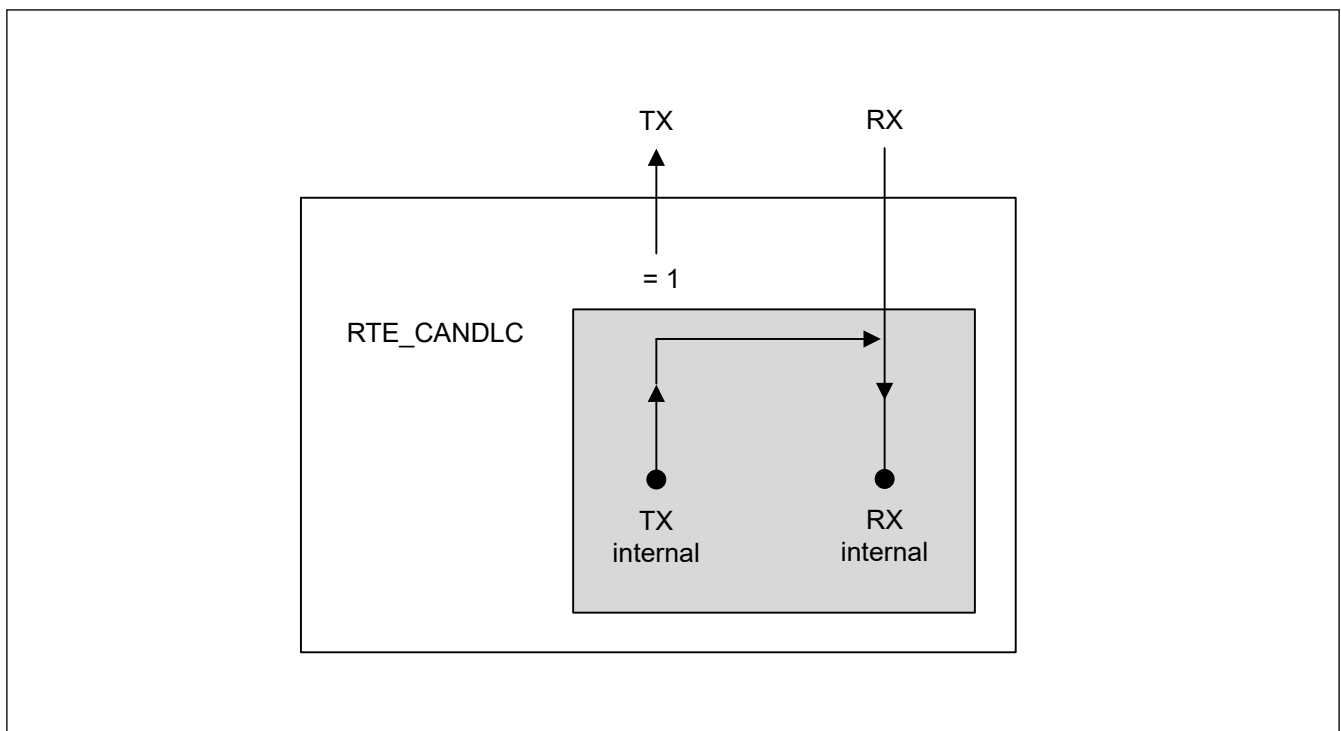
#### 32.9.1.2 Listen-only Mode

The ISO 11898-1 recommends an optional bus-monitoring mode. In this mode, the CAN channel is able to receive valid data frames and valid remote frames. However, it sends only recessive bits on the CAN bus and is not allowed to transmit.

If the CAN engine is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is routed internally so that the CAN engine monitors this as dominant. The external TX pin remains in recessive state.

This mode can be used for baud rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.

In this mode, it is not permitted to request transmission from any normal TX message buffer or TX FIFO.

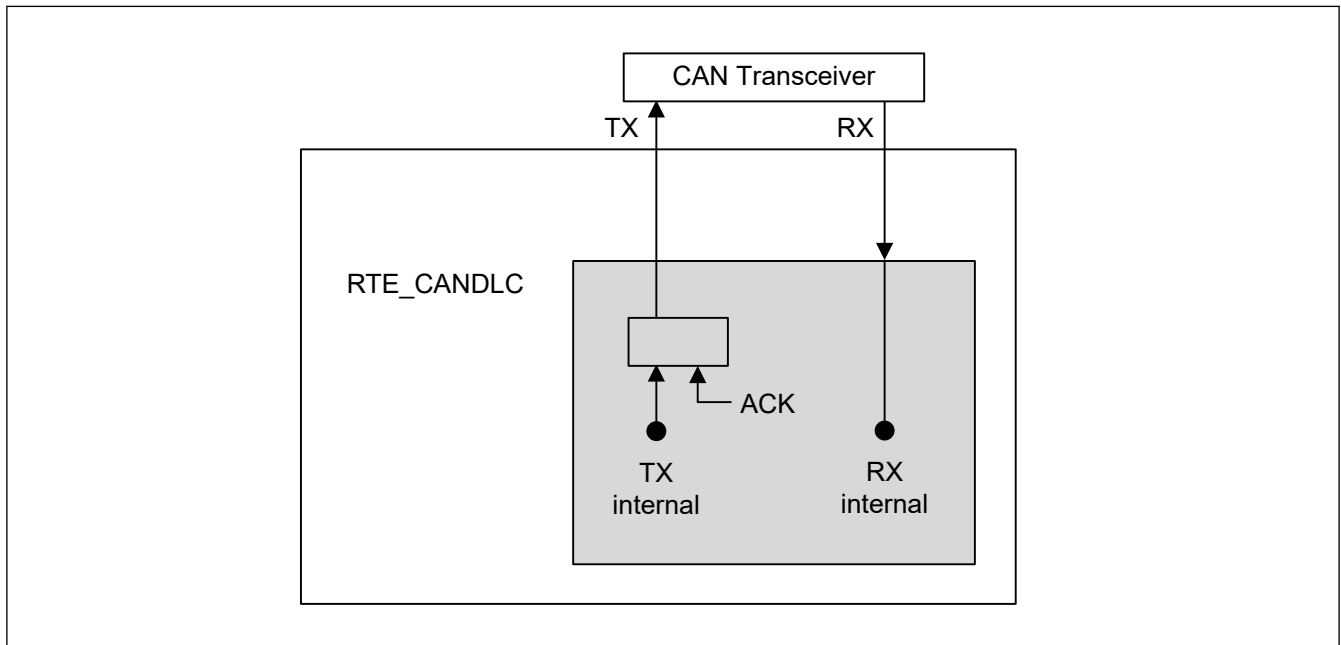


### 32.9.1.3 Self-test Mode 0 (External loopback mode)

In Self-test mode 0, the CAN engine treats its own transmitted messages as received messages through the CAN transceiver and stores them into its receive message buffers.

To be independent from external stimulation, the engine generates its own Acknowledge bit.

This test can be used for CAN transceiver tests and the RX/TX pins should be connected to the transceiver.

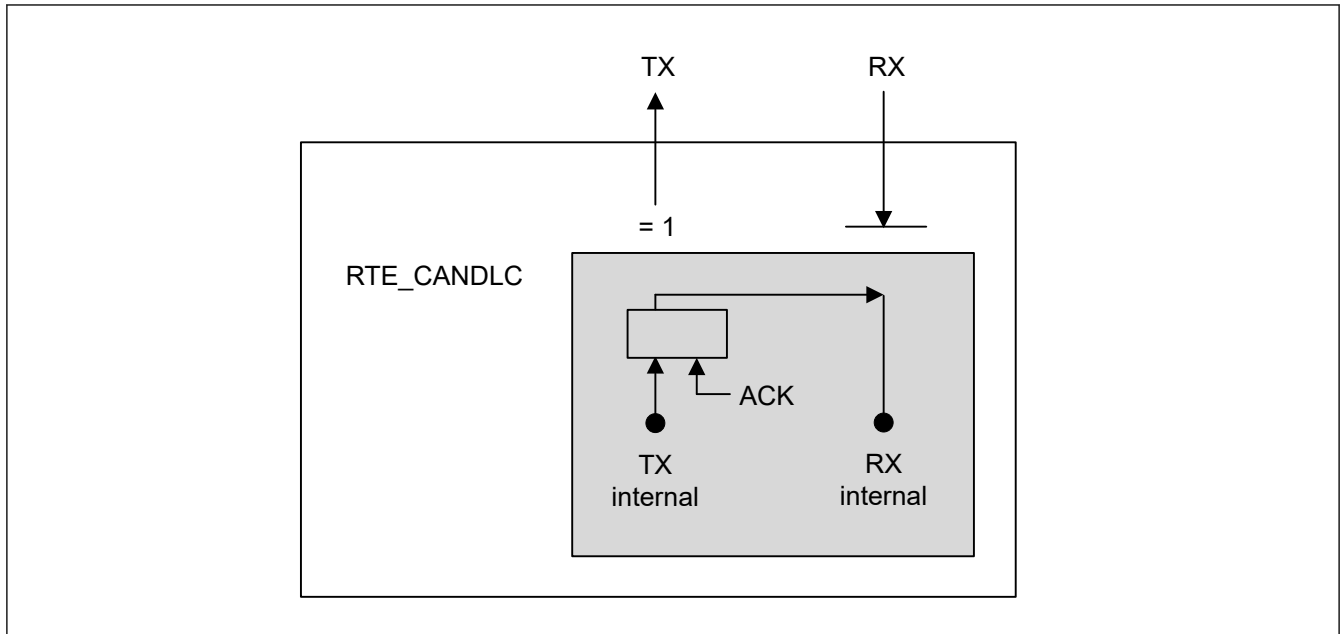


### 32.9.1.4 Self-test Mode 1 (Internal loopback mode)

In Self-test mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent from external stimulation, the CAN engine generates its own Acknowledge bit. In this mode the CAN engine performs an internal feedback from TX internal to RX internal. The actual value of the external RX input is disregarded by the CAN engine.

The external TX pin outputs only recessive bits. The RX/TX pins do not need to be connected to the CAN bus or any external device.

Note: The channel pins are also disconnected from the internal CAN bus communication line.



### 32.9.1.5 Restricted Operation Mode

This chapter is not valid for classical CAN.

In Restricted operation mode, the CAN node is able to receive valid data and remote frames generating the Acknowledge bit.

Active error or overload frames cannot be transmitted, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.

Additionally, the Receive and Transmit Error Counter (REC and TEC) are frozen independently from the occurrence of errors. The mode is specified in ISO 11898-1 and the setting of transmit request is permitted.

### 32.9.2 Global Test Modes

The CANFD module can be configured into the following test modes:

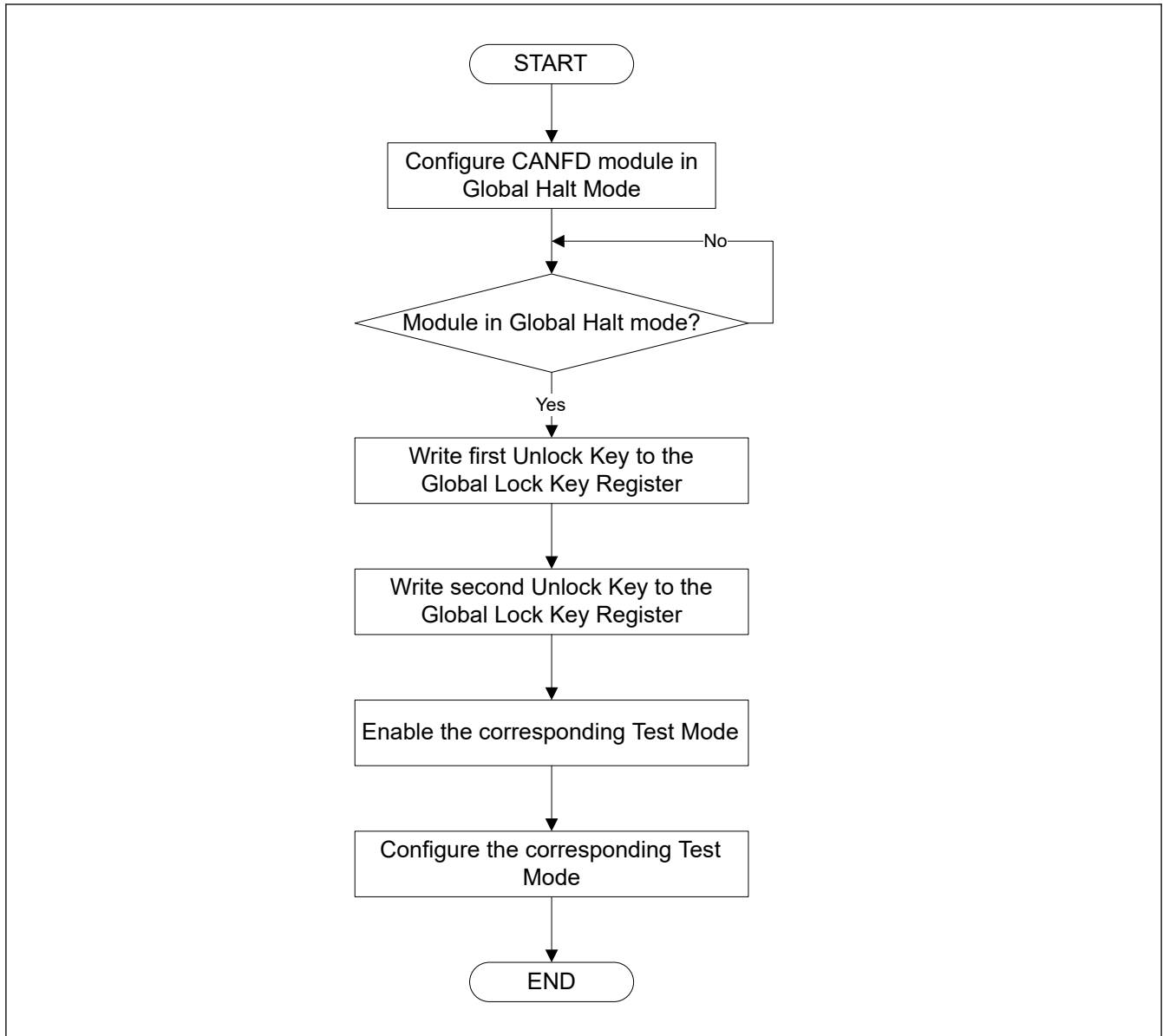
- RAM test mode
- Bit Flip Test

The test modes in the following table are protected by a special software procedure to enable the mode. This software procedure enables write access to the test mode by a specific unlock key as shown in the table.

Test mode	Unlock key 1	Unlock key 2
RAM test mode	0x7575	0x8A8A

If the software sequence of the two consecutive unlock key write accesses (half-word or word access) is interrupted by any other write access to the register or if incorrect data is written to the Global Unlock Key Register, the corresponding test mode cannot be set and the sequence must be restarted.

After the two unlock key write accesses, the next write access should be to set the corresponding test mode enable bit. If this is not followed, the unlock mechanism reset and the test mode enable bit cannot be set and the unlock sequence must be restarted.



**Figure 32.51** Unlock software protection routine

### 32.9.2.1 RAM Test Mode

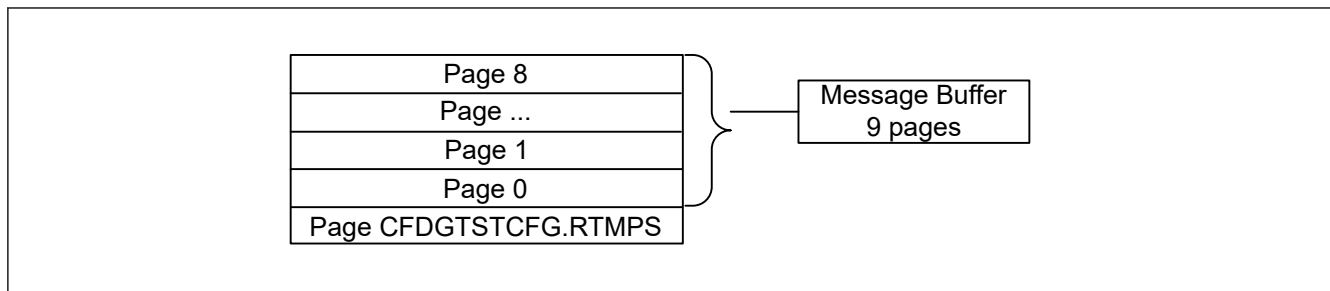
The CANFD module can be configured in RAM test mode by setting the `CFDGTSTCTR.RTME` bit in the Global Test Control Register when the corresponding lock key is previously written. This is a special test mode, in which, the complete RAM area can be accessed.

**Note:** The actual RAM size is bigger than the RAM area initialized after a hardware reset. Therefore, ECC error flag (of the ECC macro) may be set if CPU reads data from this uninitialized RAM area while CANFD module is in RAM test mode.

In this mode, the RAM area is split into number of pages (pn) of 256 bytes, each which can be accessed with the `CFDRPGACCK` register.

The page should be selected for read/write access by writing to the `CFDGTSTCFG.RTMPS[3:0]` bits in the Global Test Control Register. Data can then be read from or written in to the RAM Test Page Access Registers.

[Figure 32.52](#) shows the structure of the pages in the RAM when performing a RAM test mode.



**Figure 32.52 RAM page structure**

The total available RAM size is 2072 bytes for the Message Buffer RAM.

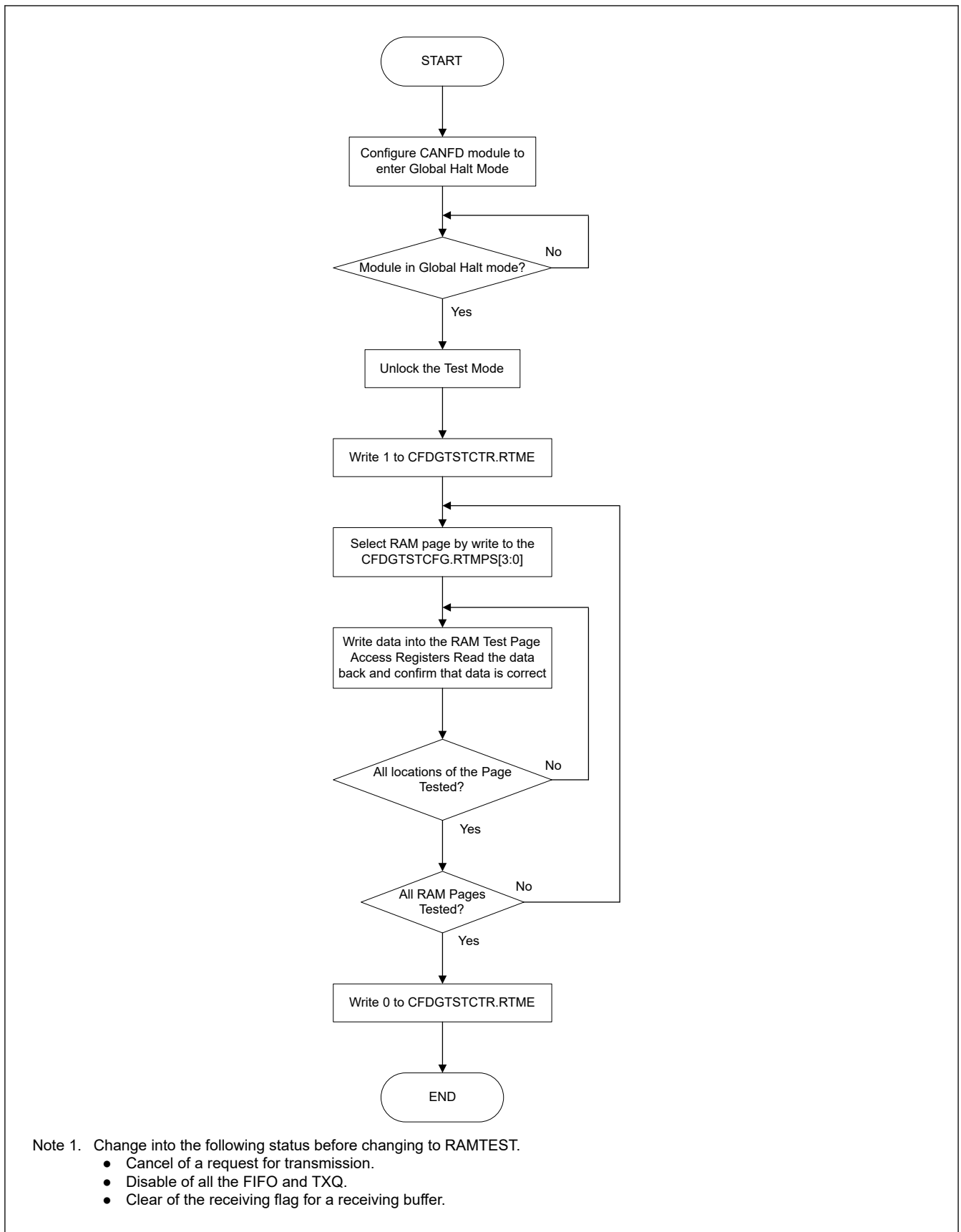
The pn and CFDGTSTCFG.RTMPS[3:0] values for the MB RAMs are calculated in the following way:

$pn = \text{ceil}(\text{total RAM size in bytes} / \text{number of bytes per page})$

- MB RAM:
  - $pn = \text{ceil}(2072 / 256) = 9 \text{ pages}$
  - $\text{CFDGTSTCFG.RTMPS}[3:0] = 0 \text{ to } 8 \text{ inclusive}$
  - (User should not access more than 24 Bytes in the last page)

Figure 32.53 shows the software flow for RAM test mode.





**Figure 32.53 Software flow for RAM test mode**

To exit this test mode, the CFDGTSTCTR.RTME bit must be cleared. The CFDGTSTCTR.RTME bit is cleared by writing 0 to it.

The CFDGTSTCTR.RTME bit is cleared automatically when the CANFD module enters Global Reset mode from the test mode.

### 32.9.2.2 Bit Flip Test

Bit Flip Test can invert the bit (the 1st bit of ID) of the beginning of the bit stream to receive.

If this function is used by a transmitting node, a bit error or an arbitration lost will occur.

If this function is used by a receiving node, a CRC error or a stuff error will occur.

Users should refer to the bit stuffing rule when using this feature, as there is the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The following sequence should be used to perform CRC Error testing. In the sequence below CANFD module is the receiver.

1. Set the CFDC0CTR.BFT bit to 1'b1, in order to invert the first bit of the incoming bit stream from sending node
2. Wait for the CANn\_CHERR (n = 0, 1) output signal to set to 1'b1
3. Read either the CFDC0ERFL.CRCREG or the CFDC0FDCRC.CRCREG (depending on the received frame type: Classical or FD). The value should be different from the received CRC value of the reference message from sending node.
4. Check that CFDC0ERFL.CERR is 1'b1

As the CRC generator logic is shared for RX and TX there is no need to create a separate TX CRC Error test.

## 32.10 RAM Area Configuration

The RAM area used in CANFD (refers to as MRAM) can be split into the following groups as shown in [Figure 32.54](#).

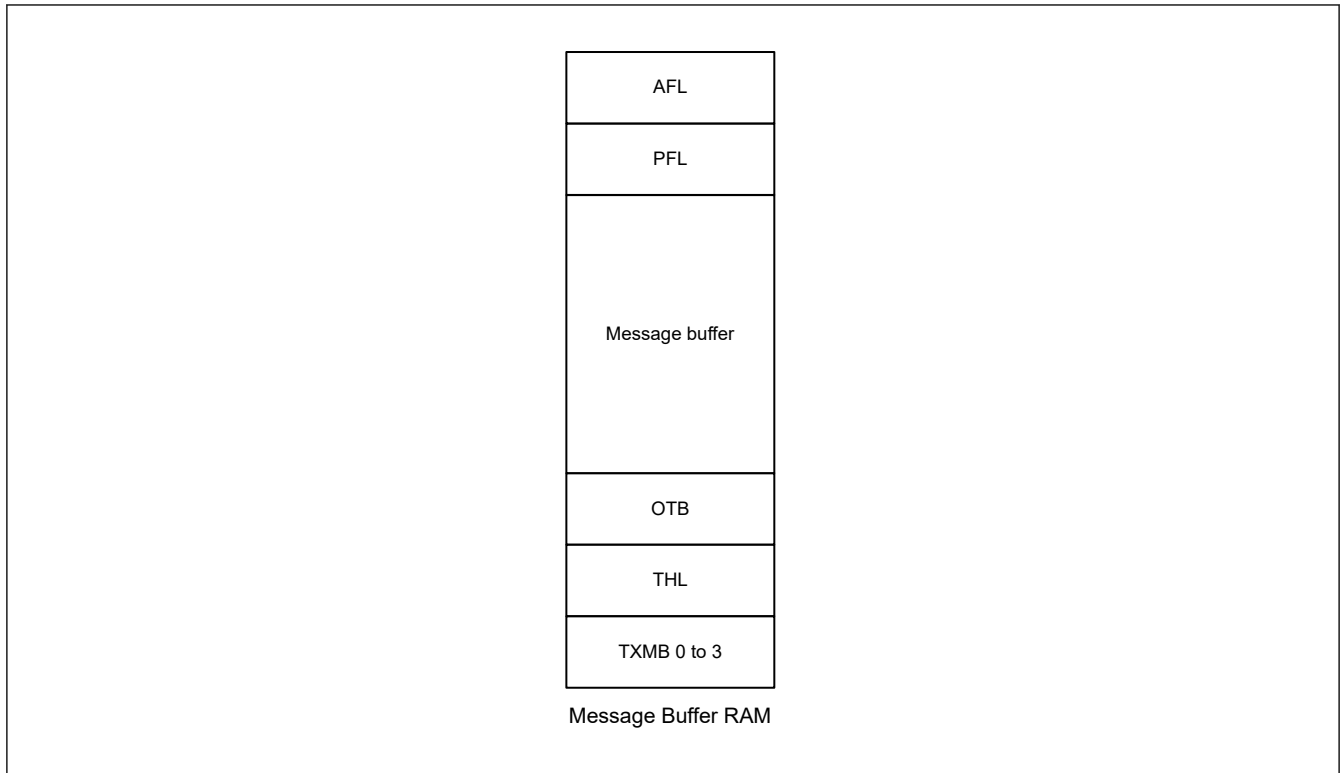
- AFL Rule Table area
- PFL Rule Table area
- Message Buffer<sup>\*1</sup> area (RX MB + FIFO Buffer)
- OTB area
- THL area
- TX MB area.

Physically, the RAM is the Message Buffer RAM<sup>\*2</sup> (RX MB, RX FIFO, Common FIFO<sup>\*3</sup> TX MB, THL, OTB, AFL Rule Table, PFL Rule Table).

Note 1. Refers to as MB

Note 2. Refers to as MRAM

Note 3. Refers to as CFIFO



**Figure 32.54 RAM area grouping**

The MRAM area starts with the TX MB area at address 0x0000. The TX MB area is followed immediately by the THL area which is then followed immediately by the OTB area. The size of the TX MB, THL, and OTB areas is fixed. The OTB area is followed by the Message Buffer area. The Message Buffer area size depends on the configuration of the flat RXMBs, RXFIFOs, and CFIFO. When all are configured, the RX MB area is followed by the RX FIFO area which is followed by the CFIFO area.

The configured MRAM area can then be calculated as follows:

$$\text{MRAM\_cfg} = \text{RXMB\_MRAM\_cfg} + \text{RXFIFO\_MRAM\_cfg} + \text{CFIFO\_MRAM\_cfg} + \text{TXMB\_MRAM\_cfg} + \text{THL\_MRAM\_cfg} + \text{OTB\_MRAM\_cfg} + \text{AFL\_MRAM\_cfg} + \text{PFL\_MRAM\_cfg}$$

$$\text{RXMB\_MRAM\_cfg} = (12 \text{ bytes} + \text{CFDRMNB.RMPLS}) \times \text{CFDRMNB.NRXMB}$$

$$\text{RXFIFO\_MRAM\_cfg} = \text{SUM} ((12 \text{ bytes} + \text{CFDRFCCa.RFPLS}) \times \text{CFDRFCCa.RFDC})$$

$$\text{CFIFO\_MRAM\_cfg} = (12 \text{ bytes} + \text{CFDCFCC.CFPLS}) \times \text{CFDCFCC.CFDC}$$

$$\text{TXMB\_MRAM\_cfg} = 304 \text{ bytes}$$

$$\text{THL\_MRAM\_cfg} = 64 \text{ bytes}$$

$$\text{OTB\_MRAM\_cfg} = 160 \text{ bytes}$$

$$\text{PFL\_MRAM\_cfg} = 72 \text{ bytes}$$

$$\text{AFL\_MRAM\_cfg} = 256 \text{ bytes}$$

“a” means RX FIFO index = [0...no\_of\_RFIFOs - 1]

no\_of\_RFIFOs : Number of configured RX FIFOs

Note: For CFDRFCCa.RFDC, CFDCFCC.CFDC, CFDRMNB.RMPLS, CFDRMNB.NRXMB, CFDRFCCa.RFPLS, and CFDCFCC.CFPLS the related number of bytes must be used.

Table 32.29 shows the calculation of the different RAM areas used for the AFL entries, OTB buffers, TX/RX message buffers, RX/Common FIFOs, and PFL entries.

**Table 32.29 MRAM area calculation**

RAM name	RAM property	RAM area calculation method	RAM values
AFL	Average rule entries	—	16
	Number of bytes in a rule entry	Fixed	16
	Number of bytes in AFL area	Average rule entries × Number of bytes in a rule entry	256
PFL	Average rule entries	—	2
	Number of bytes in a rule entry	Fixed	36
	Number of bytes in PFL area	Average rule entries × Number of bytes in a rule entry	72
TX MB	Number of TX MBs	Fixed	4
	Number of bytes required for each TX MB	Fixed	76
	Number of bytes in TX MB area	Number of TX MBs × Number of bytes required for each TX MB	304
THL	Number of entries in one THL buffer	Fixed	8
	Number of bytes required for each THL entry	Fixed	8
	Number of bytes in THL area	Number of entries in one THL buffer × Number of bytes required for each THL entry	64
OTB	Average number of buffers	—	2
	Number of bytes for OTB entry	Fixed	80
	Number of bytes in OTB area	Average number of buffers × Number of bytes for OTB entry	160
Message Buffer	Number of RX MBs	Fixed	32
	Number of RX FIFOs	Fixed	2
	Number of Common FIFO	Fixed	1
	Average number of messages for RX MB and FIFO buffers	—	16
	Number of bytes for each stored message	Fixed	—
	Average size of a Message Buffer in bytes	—	76
	Number of bytes in Message Pool area	Average number of messages for RX MB and FIFO buffers × Average size of a Message Buffer in bytes	1216
	Number of bytes message RAM	Number of bytes in Message Pool area + Number of bytes in OTB area + Number of bytes in THL area + Number of bytes in TX MB area + Number of bytes in PFL area + Number of bytes in AFL area	2072

### 32.10.1 Examples

Figure 32.55 shows one possible configuration.

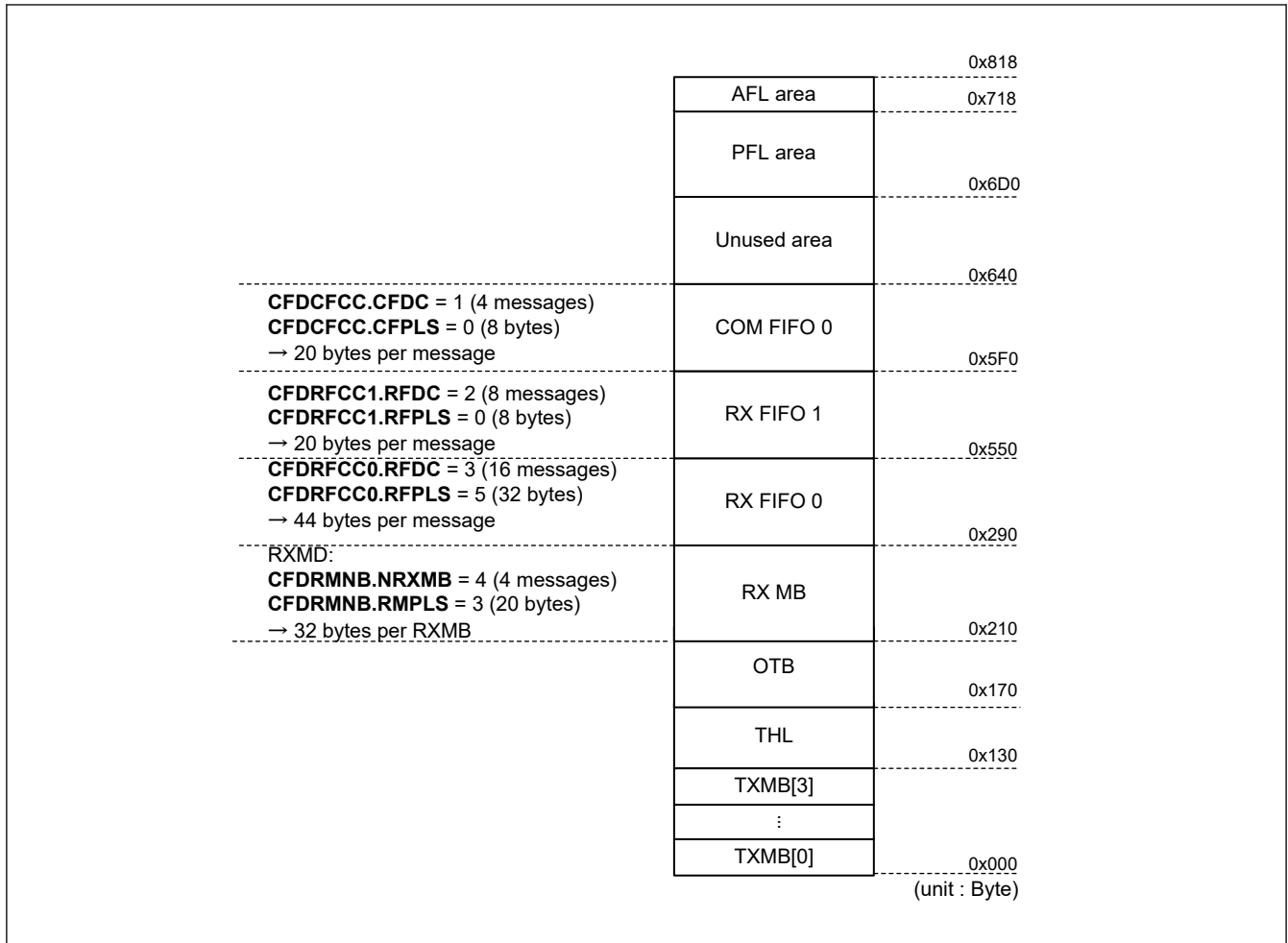


Figure 32.55 RX MB + FIFO buffers RAM area configuration examples

### 32.10.2 OTB Area

The OTB area starts immediately after the area allocated for THL buffers. The OTB is a special purpose buffer used by the CANFD. This section of RAM area can be accessed only by the CPU in RAM Test mode. Buffer requires 80 bytes and the average number of buffers is 2. Therefore, the total number of bytes allocated for the OTB is 2 × 80 bytes.

### 32.10.3 RAM Initialization Cycle

The number of RAM initialization cycles and the RAM number of pages are shown in [Table 32.30](#).

Table 32.30 RAM initialization cycle

MRAM area size	RAM initialization cycles [PCLKA cycle]	RAM Test RTMPS range
2072	520	0x0 to 0x8

## 32.11 Usage notes

### 32.11.1 Module-stop function

CANFD operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The CANFD module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

## 33. CANFD ECC (CNECC)

### 33.1 Overview

MBRAM have ECC function of 2-bit ECC error detection and 1-bit ECC error detection and correction\*1. The ECC module adds 7 bits ECC data to 32 bits RAM data.

Note 1. The ECC module cannot detect 3 or more bits error. In this case, the ECC module detects 1-bit or 2-bit error, does not detect errors, or corrects the erroneous bit to erroneous data by setting. When all RAM data are fixed to 0 or 1, it is detected as 2-bit ECC error.

### 33.2 Register Descriptions

#### 33.2.1 EC710CTL : ECC Control Register

Base address:  $ECCMBn = 0x4036\_F200 + 0x0100 \times n$  (n = 0, 1)  
 $ECCMBn\_NS = 0x5036\_F200 + 0x0100 \times n$  (n = 0, 1)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECDE DF0	ECSE DF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EMCA[1:0]	—	—	—	ECOV FF	ECER 2C	ECER 1C	—	—	ECER VF	EC1E CP	EC2E DIC	EC1E DIC	ECER 2F	ECER 1F	ECEM F
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	ECEMF	ECC Error Message Flag 0: There is no bit error in present RAM output data 1: There is bit error in present RAM output data	R
1	ECER1F	ECC Error Detection and Correction Flag 0: After clearing this bit, 1-bit error correction has not occurred 1: 1-bit error has occurred	R
2	ECER2F	2-bit ECC Error Detection Flag 0: After clearing this bit, 2-bit error has not occurred 1: 2-bit error has occurred	R
3	EC1EDIC	ECC 1-bit Error Detection Interrupt Control 0: Disable 1-bit error detection interrupt request 1: Enable 1-bit error detection interrupt request	R/W
4	EC2EDIC	ECC 2-bit Error Detection Interrupt Control 0: Disable 2-bit error detection interrupt request 1: Enable 2-bit error detection interrupt request	R/W
5	EC1ECP	ECC 1-bit Error Correction Permission 0: At 1-bit error detection, the error correction is executed 1: At 1-bit error detection, the error correction is not executed	R/W
6	ECERVF	ECC Error Judgment Enable Flag 0: Error judgment disable 1: Error judgment enable	R/W
8:7	—	These bits are read as 0. The write value should be 0.	R/W
9	ECER1C	Accumulating ECC Error Detection and Correction Flag Clear 0: No effect 1: Clear accumulating ECC error detection and correction flag	R/W
10	ECER2C	2-bit ECC Error Detection Flag Clear 0: No effect 1: Clear 2-bit ECC error detection flag	R/W

Bit	Symbol	Function	R/W
11	ECOVFF	ECC Overflow Detection Flag 0: No effect 1: ECC overflow detection flag	R
13:12	—	These bits are read as 0. The write value should be 0.	R/W
15:14	EMCA[1:0]	Access Control to ECC Mode Select bit These bits enable or disable write access to ECERVF bit.	R/W
16	ECSEDF0	ECC Single bit Error Address Detection Flag 0: There is no bit error in EC710EAD0 after reset or clearing ECER1F bit 1: Address captured in EC710EAD0 shows that 1-bit error occurred and captured	R
17	ECDEDF0	ECC Dual Bit Error Address Detection Flag 0: There is no bit error in EC710EAD0 after reset or clearing ECER2F bit 1: Address captured in EC710EAD0 shows that 2-bit error occurred and captured	R
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

### ECEMF bit (ECC Error Message Flag)

The ECEMF bit shows that there is error in present read data bus. This bit is updated by every RAM output data.

When RAM output data is undefined and the ECERVF bit is set to 1, the value of this bit is undefined.

[Setting condition]

There is bit error in present RAM output data under the condition that error judgement is enabled.

[Clearing condition]

- Under the condition that there is no 1-bit error in input data to decode circuit
- When ECC error judgement is disabled (ECERVF = 0).

### ECER1F bit (ECC Error Detection and Correction Flag)

The ECER1F bit shows that the bit errors are detected in the one part of RAM read data [38:0] at RAM read access when the error judgment is enabled.

When the 1-bit error interrupt output is enabled, error interrupt is generated by setting this flag.

This bit is read-only, so writing 1 or 0 has no effect.

At clearing, write 1 to the ECER1C bit.

When 1-bit error is detected again under the condition that this bit is set, the interrupt is not generated.

[Setting condition]

When the error judgment is enabled and there is 1-bit error to RAM output data (when not setting ECER1C = 1).

[Clearing condition]

- Writing ECER1C = 1
- When ECC error judgement is disabled (ECERVF = 0).

### ECER2F bit (2-bit ECC Error Detection Flag)

The ECER2F bit shows that the bit errors are detected in the two parts of RAM read data [38:0] at RAM read access when the error judgment is enabled.

When the 2-bit error interrupt output is enabled, error interrupt is generated by setting this flag.

This bit is read-only, so writing 1 or 0 has no effect.

At clearing, write 1 to the ECER2C bit.

When 2-bit error is detected again under the condition that this bit is set, the interrupt is not generated.

[Setting condition]

When the error judgment is enabled and there is 2-bit error to RAM output data (when not setting ECER2C = 1).

[Clearing condition]

- Writing ECER2C = 1
- When ECC error judgement is disabled (ECERVF = 0).

#### EC1EDIC bit (ECC 1-bit Error Detection Interrupt Control)

The EC1EDIC controls the interrupt output at detecting 1-bit error. By setting 1 to this bit, the 1-bit error interrupt is outputted when 1-bit error detected.

#### EC2EDIC bit (ECC 2-bit Error Detection Interrupt Control)

The EC2EDIC controls the interrupt output at detecting 2-bit error. By setting 1 to this bit, the 2-bit error interrupt is outputted when 2-bit error detected.

#### EC1ECP bit (ECC 1-bit Error Correction Permission)

The EC1ECP sets enable or disable to correct the 1-bit error when ECC error detection and correction is valid. By setting 1 to this bit, the non-corrected data is outputted if 1-bit error is detected.

#### ECERVF bit (ECC Error Judgment Enable Flag)

Setting the ECERVF bit to 1 enables the judgment of error. The correction of output data and the interrupt output depend on setting of the EC1ECP bit, EC2EDIC bit, and EC1EDIC bit.

The write access to this bit is valid when the write value of the EMCA[1:0] is 01b. So only the 16 bits or 32 bits operation command is valid in the case of the write access to this bit.

#### ECER1C bit (Accumulating ECC Error Detection and Correction Flag Clear)

The ECER1C bit clears the status flag of the ECER1F bit.

The read value is always 0. By writing 0, the internal condition is not changed. When the competition between writing 1 to this bit and setting the ECER1F bit, the former has priority.

The ECER1F bit is cleared by writing 1 to this bit while the ECER1F bit is set. Additionally, the Overflow Detection flag (ECOVFF), ECC Dual Bit Error flag (ECDEDF0) and ECC Single Bit Error flag (ECSEDF0) are also cleared.

#### ECER2C bit (2-bit ECC Error Detection Flag Clear)

The ECER2C bit clears the status flag of the ECER2F bit.

The read value is always 0. By writing 0, the internal condition is not changed. When the competition between writing 1 to this bit and setting the ECER2F bit, the former has priority.

The ECER2F bit is cleared by writing 1 to this bit while the ECER2F bit is set. Additionally, the Overflow Detection flag (ECOVFF), ECC Dual Bit Error flag (ECDEDF0), and ECC Single Bit Error flag (ECSEDF0) are also cleared.

#### ECOVFF bit (ECC Overflow Detection Flag)

The ECOVFF bit is set and the overflow interruption is outputted by detecting the new error address under the condition that error address is already captured in the EC710EAD0 register. The overflow interrupt is outputted again when this bit is set and new error is detected.

This bit is read-only, so writing 1 or 0 has no effect.

To clear this bit, write 1 to the ECER2C bit and the ECER1C bit.

[Setting condition]

When new error address is captured under the condition that error address is already captured in the EC710EAD0 register (when not setting ECER2C = 1 or ECER1C = 1).

[Clearing condition]

- Writing ECER2C = 1 or ECER1C = 1
- When ECC error judgement is disabled (ECERVF = 0).

#### EMCA[1:0] bit (Access Control to ECC Mode Select bit)

The EMCA[1:0] bits are the write trigger reserved bits to the ECERVF bit. The read value is always 0. When the value of these bits is 01b, it is possible to have write access to the ECERVF bit. If these bits are not 01b, write access to the ECERVF bit is ignored and the value is not written.



**ECSEDF0 bit (ECC Single bit Error Address Detection Flag)**

The ECSEDF0 bit shows that the error is captured in the error address register when error detection is valid. This bit is set by 1-bit error detection.

When 1-bit error is detected after the 2-bit error address is already captured in the EC710EAD0 register, this bit is not updated but the EC710EAD0 register is updated.

This bit is read-only, so writing 1 or 0 has no effect. To clear these bits, write 1 to the ECER1C bit.

[Setting condition]

When there is 1-bit error to RAM output data and error address is captured in EC710EAD0 under the condition that the error judgment is permitted (when not setting ECER1C = 1).

[Clearing condition]

- Writing ECER1C = 1
- When ECC error judgement is disabled (ECERVF = 0).

**ECDEDF0 bit (ECC Dual Bit Error Address Detection Flag)**

The ECDEDF0 bit shows that the error is captured in the error address register when error detection is valid. This bit is set by 2-bit error detection.

When 2-bit error is detected after the 1-bit error address is already captured in the EC710EAD0 register, this bit is not updated and the EC710EAD0 register is updated.

This bit is read-only, so writing 1 or 0 has no effect. To clear these bits, write 1 to the ECER2C bit.

[Setting condition]

When there is 2-bit error to RAM output data and error address is captured in EC710EAD0 under the condition that the error judgment is permitted (when not setting ECER2C = 1).

[Clearing condition]

- Writing ECER2C = 1
- When ECC error judgement is disabled (ECERVF = 0).

**33.2.2 EC710TMC : ECC Test Mode Control Register**

Base address: ECCMBn = 0x4036\_F200 + 0x0100 × n (n = 0, 1)  
ECCMBn\_NS = 0x5036\_F200 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ETMA[1:0]	—	—	—	—	—	—	ECTM CE	—	—	—	—	—	—	ECDC S	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ECDCS	ECC Decode Input Select 0: Input lower 32 bits of RAM output data to data area of decode circuit 1: Input ECEDB31-0 in EC710TED register to data area of decode circuit	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	ECTMCE	ECC Test Mode Control Enable 0: The access to test mode register and bit is disabled 1: The access to test mode register and bit is enabled	R/W
13:8	—	These bits are read as 0. The write value should be 0.	R/W
15:14	ETMA[1:0]	ECC Test Mode Bit Access Control These bits enable or disable write access to ECTMCE bit.	R/W

Note: S-TYPE-3, P-TYPE-3

**ECDCS bit (ECC Decode Input Select)**

The ECDCS bit selects either the lower 32 bits data value from RAM or value from the internal test register (EDEDDB[31:0] in EC710TED) as input signal to decoder.

The write access to this bit is valid under the condition of ECTMCE = 1 (it is possible to set them at the same time.)

This bit is cleared by setting ECTMCE = 0.

**ECTMCE bit (ECC Test Mode Control Enable)**

The ECTMCE bit selects the access enable or disable to test register and test control bit.

The write access to this bit is valid under the condition that the value of the ETMA[1:0] bits is 10b.

**ETMA[1:0] bits (ECC Test Mode Bit Access Control)**

The ETMA[1:0] bits are the write trigger reserved bits to the ECTMCE bit. The read value is always 0. When the value of these bits is 10b, it is possible to have write access to the ECTMCE bit. If these bits are not 10b, the write access to the ECTMCE bit is ignored and the value is not written.

**33.2.3 EC710TED : ECC Test Substitute Data Register**

Base address: ECCMBn = 0x4036\_F200 + 0x0100 × n (n = 0, 1)  
 ECCMBn\_NS = 0x5036\_F200 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	ECEDB[31:0]	ECC Test Substitute Data Substitute data in ECC test mode.	R/W

Note: S-TYPE-3, P-TYPE-3

This register is for the 32 bits data for ECC decode. It is possible to read and write using 32 bits operation command when ECTMCE = 1. When ECTMCE = 0, all bits are always 0.

**ECEDB[31:0] bits (ECC Test Substitute Data)**

When ECDCS in EC710TMC register is 1, the value of this register is bits [31:0] of the input data to the decode circuit.

**33.2.4 EC710EAD0 : ECC Error Address Register**

Base address: ECCMBn = 0x4036\_F200 + 0x0100 × n (n = 0, 1)  
 ECCMBn\_NS = 0x5036\_F200 + 0x0100 × n (n = 0, 1)

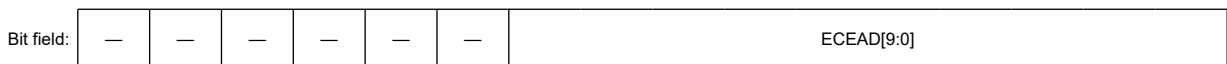
Offset address: 0x10

Bit position: 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
9:0	ECEAD[9:0]	ECC Error Address	R

Bit	Symbol	Function	R/W
31:10	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

This is a read-only register to hold the ECC error address.

### ECEAD[9:0] bits (ECC Error Address)

When ECC error is detected for permitting ECC error judgment, RAM address is captured by the detected signal as a trigger and is hold as the error occurring address. The error address is not captured when the error occurred again to the one held by the same factor.

If 2-bit error occurred under the condition that 1-bit error address is already captured, the 2-bit error address is over-written and the ECDEDF0 bit is set to 1.

If 1-bit error occurred under the condition that 2-bit error address is already captured, the 1-bit error address is not overwritten and the ECSEDF0 bit is not set to 1.

## 33.3 Operation

### 33.3.1 ECC Function Setting

Figure 33.1 shows a procedure for ECC function setting.

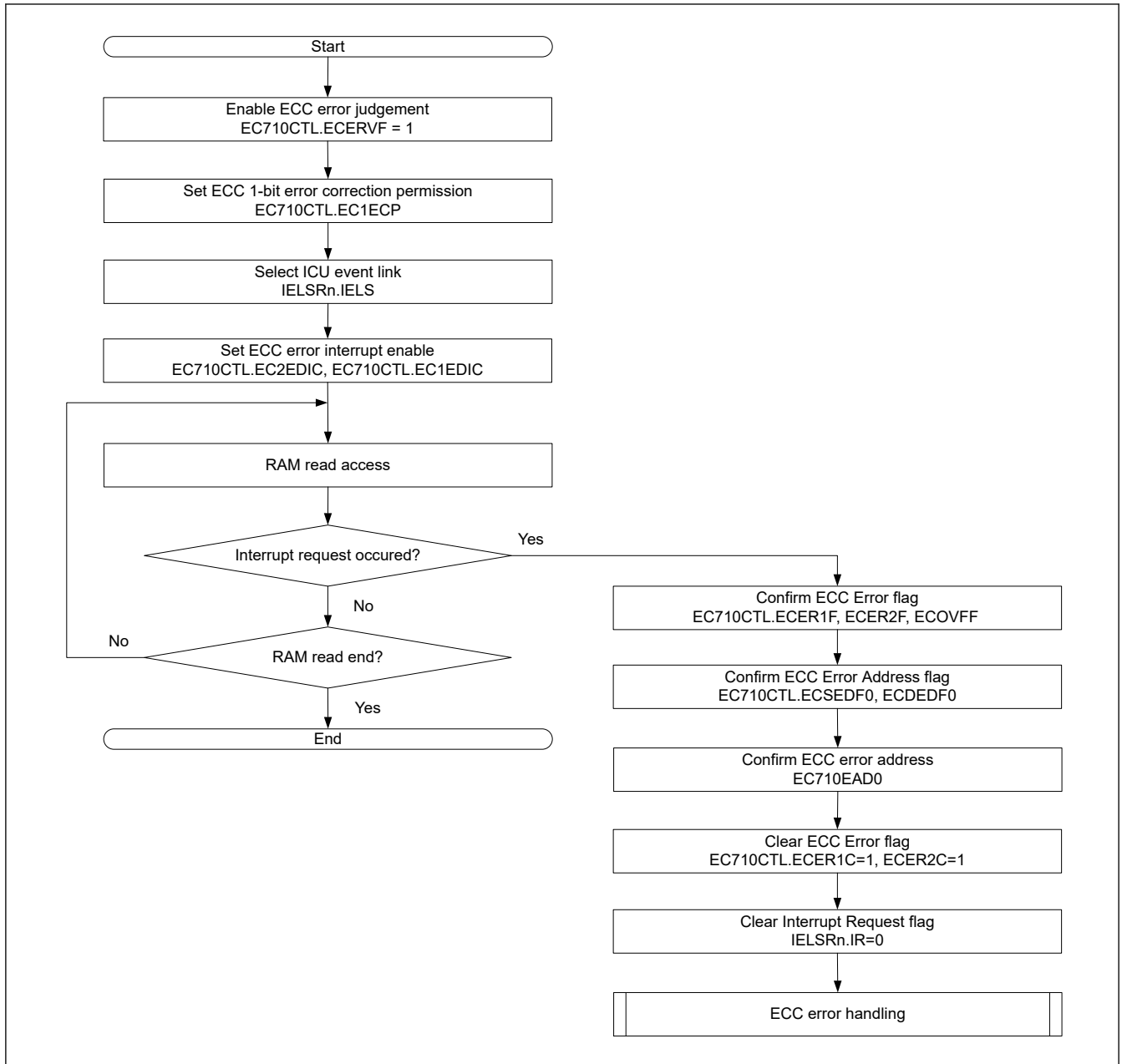


Figure 33.1 Setting procedure for ECC function

### 33.3.2 ECC Decoder Testing

ECC interrupts can be intentionally generated by ECC test mode. [Figure 33.2](#) shows a procedure for ECC decoder testing.

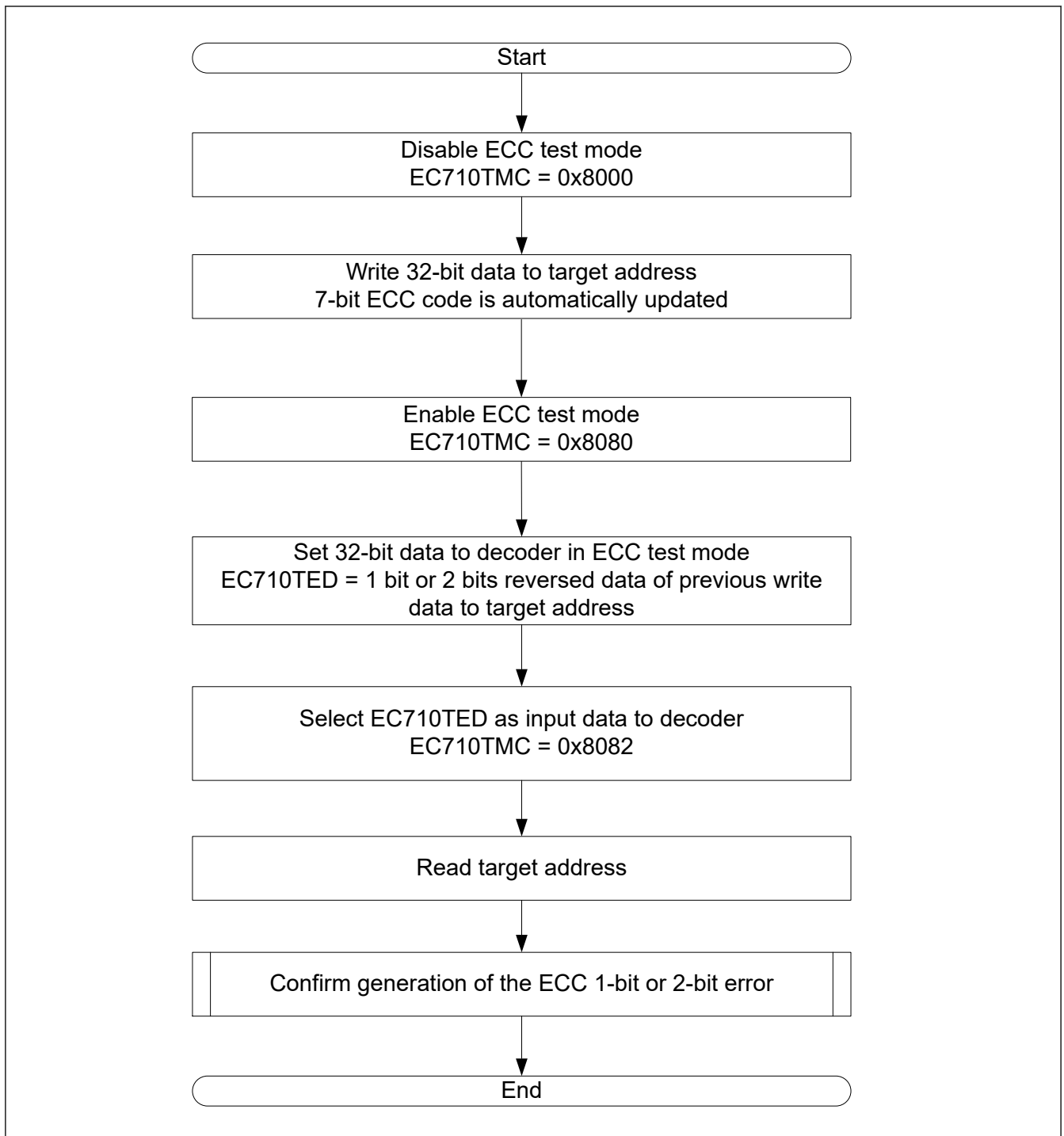


Figure 33.2 Testing procedure for ECC decoder

### 33.4 Interrupts

The ECC module issues three interrupt requests:

- CAN<sub>n</sub>\_MRAM\_ERI (n = 0, 1)

Interrupt sources of each interrupt request include:

- 1-bit ECC error
- 2-bit ECC error
- ECC error overflow.

## 34. Serial Peripheral Interface (SPI)

This is the SPI\_B version of the SPI peripheral module.

SPI\_B is referred to as SPI in this chapter.

### 34.1 Overview

The Serial Peripheral Interface (SPI) has 2 channels. The SPI provides high-speed full-duplex synchronous serial communications with multiple processors and peripheral devices. [Table 34.1](#) lists the SPI specifications, [Figure 34.1](#) shows a block diagram of SPI, [Figure 34.2](#) shows a clock source selector block diagram and [Table 34.2](#) lists the I/O pins.

In this section, PCLK refers to PCLKA.

**Table 34.1 SPI specifications (1 of 2)**

Parameter	Specifications
Number of channels	Two channels
SPI transfer functions	<ul style="list-style-type: none"> <li>Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (SPI clock) signals allows serial communications through SPI operation (4-wire method) or clock synchronous operation (3-wire method)</li> <li>Transmit-only operation available</li> <li>Receive-only operation is available</li> <li>Communication mode selectable to full-duplex, transmit-only or receive-only</li> <li>RSPCK polarity switching</li> <li>RSPCK phase switching</li> </ul>
Data format	<ul style="list-style-type: none"> <li>MSB-first or LSB-first selectable</li> <li>Transfer bit length selectable from 4 to 32 bits</li> <li>32 bit × 4 stages FIFO is available as transmit buffer or receive buffer</li> <li>Byte swap operating function</li> <li>Transmit/receive data can be inverted.</li> </ul>
Operation clock (TCLK)	Synchronized clock (PCLK) or independent clock (SPICLK) can be selected.
Bit rate	<ul style="list-style-type: none"> <li>In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing TCLK (the division ratio ranges from divided by 2 to divided by 4096)</li> <li>In slave mode, the minimum TCLK clock divided by 2 can be input as RSPCK (TCLK divided by 2 is the maximum RSPCK frequency)</li> </ul> Width at high level: 1 TCLK cycle; width at low level: 1 TCLK cycle
Buffer configuration	<ul style="list-style-type: none"> <li>Double buffer configuration for the transmit and receive buffers</li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Mode fault error detection</li> <li>Underrun error detection</li> <li>Overrun error detection*1</li> <li>Parity error detection</li> <li>Receive data ready detection</li> </ul>
SSL control function	[motorola SPI mode/TI SSP mode common] <ul style="list-style-type: none"> <li>Four SSL pins (SSLn: SSLn0 to SSLn3) (n = A, B) for each channel</li> <li>In single-master mode, SSLn0 to SSLn3 pins are output</li> <li>In multi-master mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins either for output or unused</li> <li>In slave mode, SSLn0 pin for input and SSLn1 to SSLn3 pins unused</li> <li>Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)</li> <li>Function for changing SSL polarity</li> <li>Delay between frames in burst transfer is settable</li> </ul> [only Motorola mode] <ul style="list-style-type: none"> <li>Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)</li> <li>Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)</li> </ul> [only TI SSP mode] <ul style="list-style-type: none"> <li>Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 0 to 8 RSPCK cycles (set in RSPCK-cycle units)</li> <li>Controllable delay from OE output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)</li> </ul>
Communication protocol	<ul style="list-style-type: none"> <li>Motorola SPI</li> <li>TI SSP(Synchronous Serial Protocol)</li> </ul>

**Table 34.1 SPI specifications (2 of 2)**

Parameter	Specifications
Synchronization bypass function	Synchronization circuit can be bypassed only when the same clock is input to bus clock (PCLK) and operation clock (TCLK).
Control in master transfer	<ul style="list-style-type: none"> <li>• Transfers of up to eight commands each can be executed sequentially in looped execution</li> <li>• For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, MSB- or LSB-first, burst, RSPCK delay, SSL negation delay, and next-access delay</li> <li>• Transfers can be initiated by writing to the transmit buffer</li> <li>• MOSI signal value specifiable in SSL negation</li> <li>• RSPCK auto-stop function</li> </ul>
Interrupt sources	Interrupt sources: <ul style="list-style-type: none"> <li>• Receive buffer full / Receive data ready interrupt</li> <li>• Transmit buffer empty interrupt</li> <li>• SPI error interrupt (mode fault error, under run error, over run error, parity error, receive data ready)</li> <li>• SPI idle interrupt (SPI idle)</li> <li>• Communication end interrupt</li> </ul>
Event link function	The following events can be output to the Event Link Controller (ELC): <ul style="list-style-type: none"> <li>• Receive buffer full / receive data ready signal</li> <li>• Transmit buffer empty signal</li> <li>• Mode fault, underrun, overrun, parity error, or receive data ready signal</li> <li>• SPI idle signal</li> <li>• Communication end signal</li> </ul>
Other functions	<ul style="list-style-type: none"> <li>• Switching between CMOS output and open-drain output</li> <li>• SPI initialization function</li> <li>• Loopback mode</li> <li>• SPE bit status polling function</li> </ul>
Module-stop function	Module-stop state can be set for each channels to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set for each channels.

Note 1. In master reception and when the RSPCK auto-stop function is enabled, an overrun error does not occur because the transfer clock is stopped on overrun error detection.

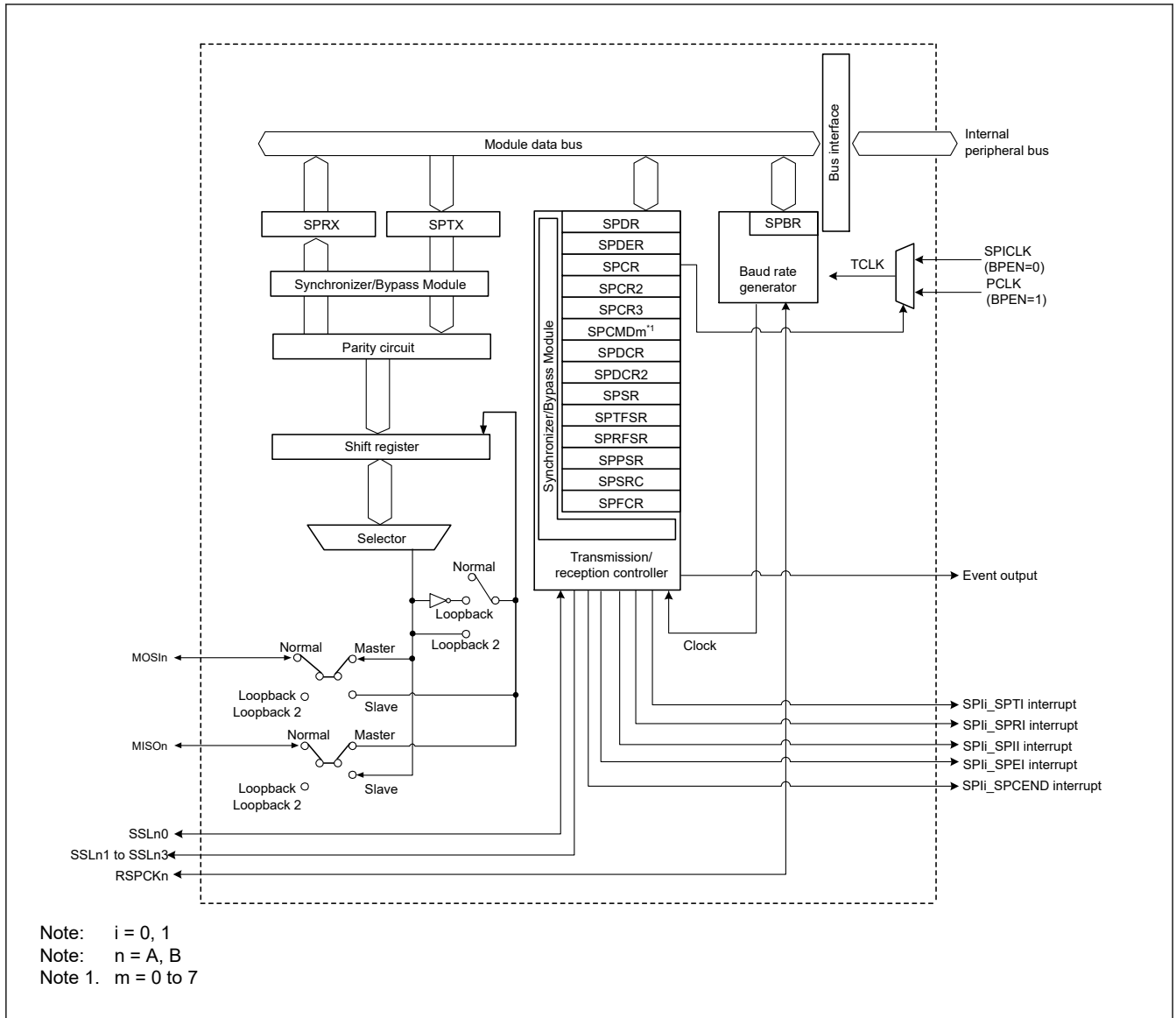
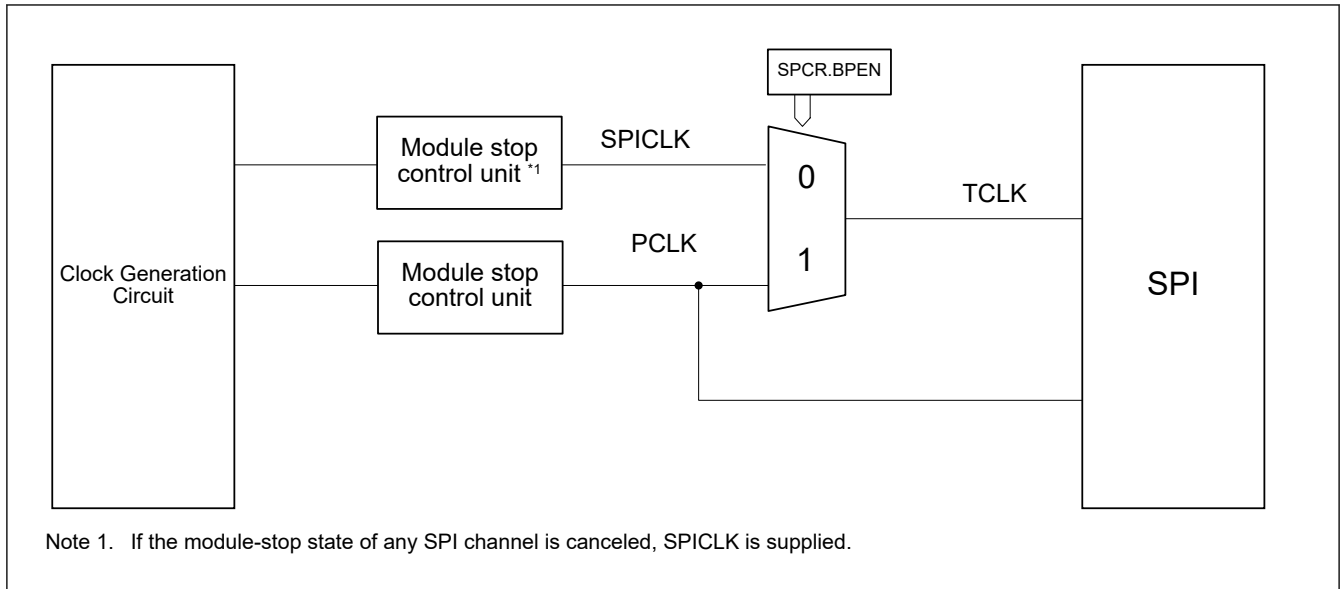


Figure 34.1 SPI block diagram





**Figure 34.2** Clock source selector block diagram

The SPI automatically switches the I/O direction of the SSLn0 pin. SSLn0 is set as an output when the SPI is a single master, and as an input when the SPI is a multi-master or a slave. The RSPCKn, MOSIn, and MISO<sub>n</sub> pins are automatically set as inputs or outputs based on the master or slave setting and the level input on the SSLn0 pin. For details, see [section 34.3.2. Controlling the SPI Pins](#).

**Table 34.2** SPI I/O pins

Channel	Pin name	I/O	Description
SPI0	RSPCKA	I/O	Clock input/output pin
	SSLA0	I/O	Slave selection input/output
	SSLA1 to SSLA3	Output	Slave selection output
	MOSIA	I/O	Master transmit data input/output
	MISOA	I/O	Slave transmit data input/output
SPI1	RSPCKB	I/O	Clock input/output pin
	SSLB0	I/O	Slave selection input/output
	SSLB1 to SSLB3	Output	Slave selection output
	MOSIB	I/O	Master transmit data input/output
	MISOB	I/O	Slave transmit data input/output

Note: Pin names are indicated as "...A" or "...An" for SPI0, and "...B" or "...Bn" for SPI1 (n = 0, 1, 2, or 3).

## 34.2 Register Descriptions

### 34.2.1 SPDR : SPI Data Register

Base address:  $SPIn\_B = 0x4035\_C000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $SPIn\_B\_NS = 0x5035\_C000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SPD[31:16]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SPD[31:0]	These bits are the interface with the buffers that hold data for transmission and reception by the SPI.	R/W

Note: S-TYPE-3, P-TYPE-3

SPDR is the interface with the buffers that hold data for transmission and reception by the SPI. When accessing this register in words, access SPD<sub>R</sub>. The transmit buffer (SPTX) and receive buffer (SPRX) are independent but both are mapped to SPD<sub>R</sub>. Figure 34.3 shows the configuration of the SPD<sub>R</sub> register.

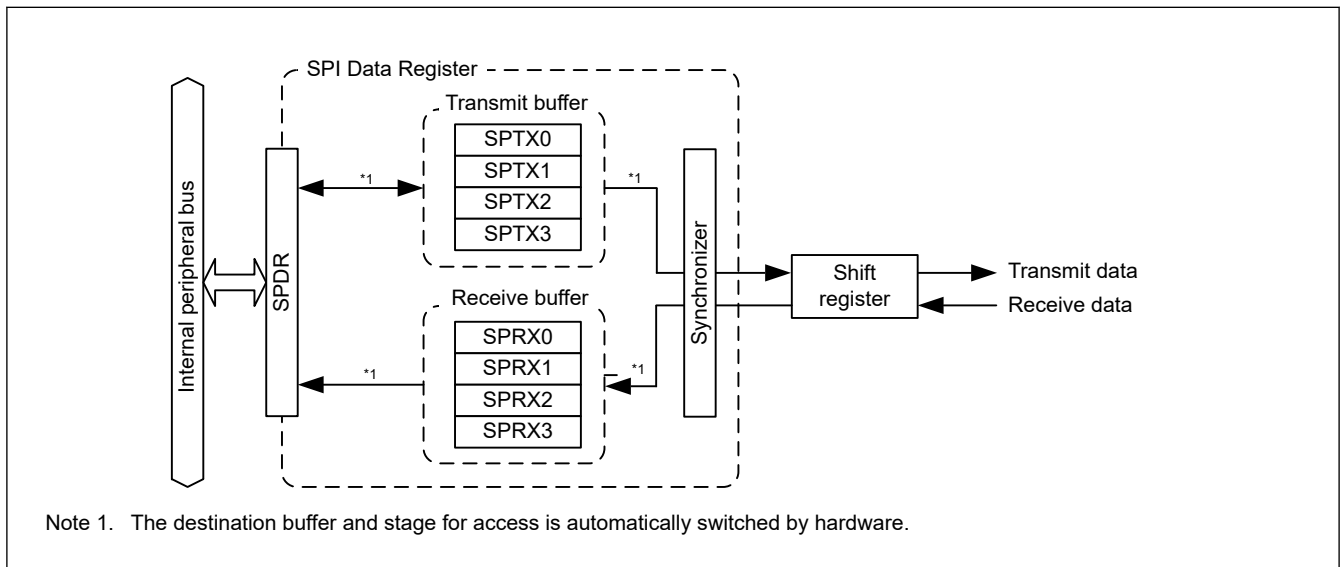


Figure 34.3 Structure of SPDR

32 bits × 4 stage transmit FIFO and 32 bits × 4 stage receive FIFO are provided. These 8 stage FIFO are mapped to one address in the SPD<sub>R</sub>. Transmit buffers (SPTX<sub>n</sub>, n = 0 to 3) can be written by writing data to SPD<sub>R</sub> to transmit written data.

Upon completion of receiving data, receive buffers store received data. When an overrun error occurs, data in the receive buffer is not updated.

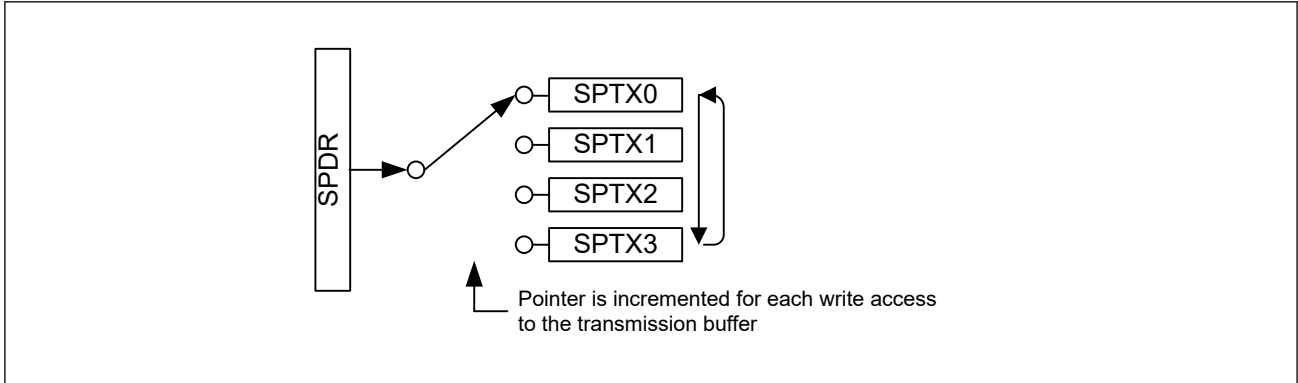
#### (1) Bus Interface

The SPI data register has 32 bits × 4 stage transmit FIFO and 32 bits × 4 stage receive FIFO (32 bytes in total). These 32 bytes are mapped to the 4-byte space of SPD<sub>R</sub>. Write transmit data from the LSB. Received data is stored from the LSB.

SPD<sub>R</sub> register write operation and read operation are described below.

##### 1. Write

A transmit buffer write pointer is provided for transmit buffers. When data is written to SPDR, the pointer automatically switches to the next buffer. The following illustrates the structure of the transmit buffer bus interface (write).



**Figure 34.4 Structure of SPDR (Write)**

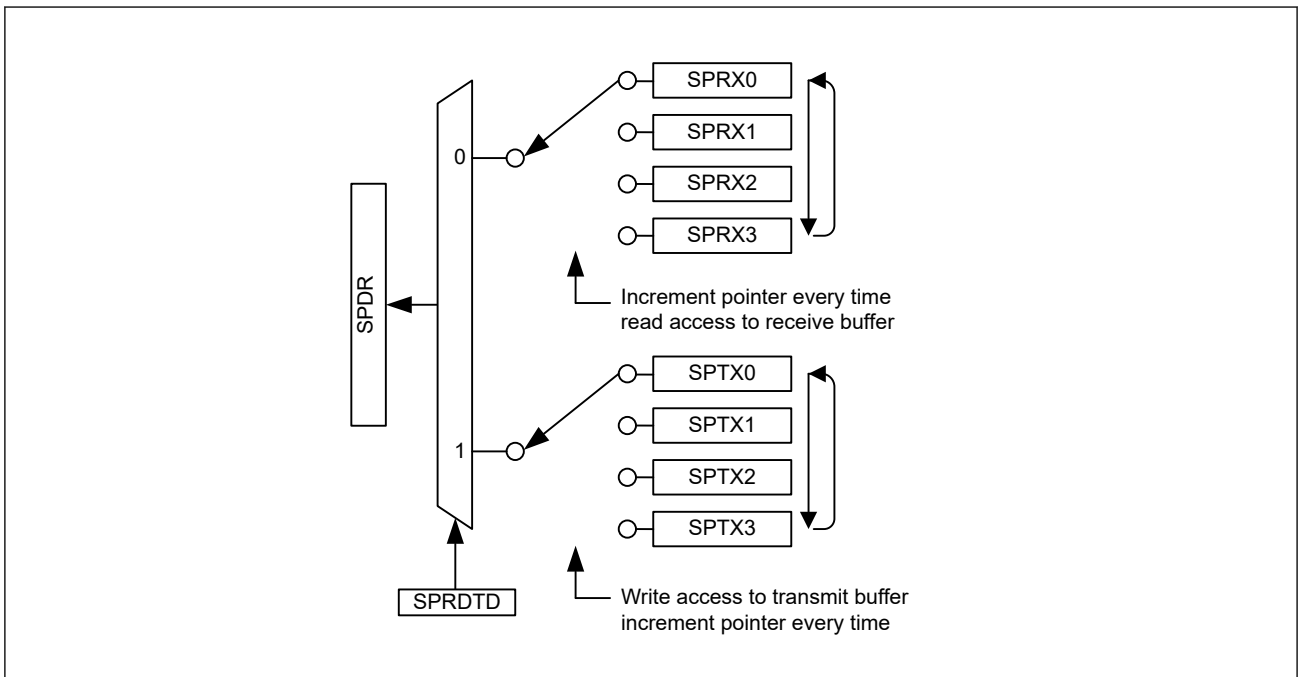
The transmit buffer (SPTX0 to SPTX3) switching order:  
 SPTX0→SPTX1→SPTX2→SPTX3→SPTX0→SPTX1→...

When writing transmit data to transmit buffers (SPTX<sub>n</sub>), write transmit data of frames +1 specified by the Transmission FIFO threshold setting bits of SPI data control register 2 (SPDCR2.TTRG[1:0]) while an SPI transmit buffer empty interrupt is present (SPSR.SPTEF flag = 1). Writing to the transmit buffer (SPTX<sub>n</sub>, n = 0 to 3) in the state where there is no empty stage in the transmit FIFO does not update the buffer value.

2. Read

Values can be read from receive buffers (SPRX<sub>n</sub>, n = 0 to 3) or transmit buffers (SPTX<sub>n</sub>, n = 0 to 3) by reading the SPDR register. Reading a receive buffer or reading a transmit buffer can be selected by the SPI receive data or transmit data select bit (SPDCR.SPRDTD) in the SPI data control register.

The SPDR register is read according to the independent receive buffer read pointer and the transmit buffer read pointer. The following illustrates the structure of the receive buffer and transmit buffer bus interface (read).



**Figure 34.5 Structure of SPDR (Read)**

When a receive buffer is read, the receive buffer read pointer automatically switches to the next buffer. The receive buffer read pointer switches in the same order as the transmit buffer write pointer. The transmit buffer read pointer is updated during the SPDR write access, but it is not updated during the transmit buffer read access. When a transmit buffer is read, the value written to SPDR last can be read.

### 34.2.2 SPDECR : SPI Delay Control Register

Base address: SPIn\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
 SPIn\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SPNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SLNDL[2:0]			—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SCKDL[2:0]	RSPCK Delay 0 0 0: 1RSPCK 0 0 1: 2RSPCK 0 1 0: 3RSPCK 0 1 1: 4RSPCK 1 0 0: 5RSPCK 1 0 1: 6RSPCK 1 1 0: 7RSPCK 1 1 1: 8RSPCK	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
10:8	SLNDL[2:0]	SSL Negation Delay [Master Mode] 0 0 0: 1RSPCK 0 0 1: 2RSPCK 0 1 0: 3RSPCK 0 1 1: 4RSPCK 1 0 0: 5RSPCK 1 0 1: 6RSPCK 1 1 0: 7RSPCK 1 1 1: 8RSPCK [Motorola-SPI case in Slave Mode] 0 0 0: 1RSPCK Others: Setting prohibited [TI-SSP case in Slave Mode] 0 0 0: 1 TCLK 0 0 1: 2 TCLK 0 1 0: 3 TCLK 0 1 1: 4 TCLK 1 0 0: 5 TCLK 1 0 1: 6 TCLK 1 1 0: 7 TCLK 1 1 1: 8 TCLK	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	SPNDL[2:0]	SPI Next-Access Delay 0 0 0: 1RSPCK + 5TCLK 0 0 1: 2RSPCK + 5TCLK 0 1 0: 3RSPCK + 5TCLK 0 1 1: 4RSPCK + 5TCLK 1 0 0: 5RSPCK + 5TCLK 1 0 1: 6RSPCK + 5TCLK 1 1 0: 7RSPCK + 5TCLK 1 1 1: 8RSPCK + 5TCLK	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

**SCKDL[2:0] bit (RSPCK Delay)**

[In the Motorola-SPI case]

The RSPCK delay bits (SCKDL) are used to set the period (RSPCK delay) from SSL signal assertion start until RSPCK oscillates while the SCKDEN bit in the SPI command register (SPCMD) is 1. If SCKDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

To use the SPI in slave mode, set SCKDL[2:0] bits to 000b.

[In the TI-SSP case]

The RSPCK delay bits (SCKDL) are used to set the period (RSPCK delay) from SSL signal assertion start until RSPCK oscillates while the SCKDEN bit in the SPI command register (SPCMD) is 1. Also that is used to set the period until the SSL signal is negated. If SCKDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

To use the SPI in slave mode, set SCKDL[2:0] bits to 000b.

**SLNDL[2:0] bit (SSL Negation Delay)**

[In the Motorola-SPI case]

The SSL negation delay bits (SLNDL) are used to set the period (SSL negation delay) after the SPI in master mode sends the final RSPCK edge during serial transfer until it negates the SSL signal while the SLNDEN bit in the SPI command register (SPCMD) is 1. If SLNDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

To use the SPI in slave mode except TI-SSP, set SLNDL[2:0] bits to 000b.

[In the TI-SSP case]

The SSL negation delay bits (SLNDL) are used to set the period (OE negation delay) after the SPI in master mode sends the final RSPCK edge during serial transfer until it negates the OE signal while the SLNDEN bit in the SPI command register (SPCMD) is 1. Also, that is used to set the period from when the SPI in slave mode detects the last RSPCK edge of serial transfer to when the OE signal is negated. If SLNDL is modified while the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

**SPNDL[2:0] bit (SPI Next-Access Delay)**

The SPI next-access delay register (SPDECR.SPNDL) is used to set the SSL signal inactive period (next-access delay) after completion of serial transfer while the SPNDEN bit in the SPI command register (SPCMD) is 1. If SPNDL is modified while the MSTR bit and the SPE bit in the SPI control register (SPCR) are 1, subsequent operation is not guaranteed.

These bits are used to set the next-access delay value when the SPNDEN bit in SPCMD is 1. To use the SPI in slave mode, set SPNDL[2:0] bits to 000b.

**34.2.3 SPCR : SPI Control Register**

Base address: SPIn\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
SPIn\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BPEN	MSTR	TXMD[1:0]	—	—	SPFR F	SPMS	—	—	CENDI E	SPTIE	SPDR ES	SPIIE	SPRIE	SPEIE	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MODF EN	BFDS	SCKA SE	PTE	—	SPOE	SPPE	—	—	—	—	—	—	—	SPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPE	SPI Function Enable 0: SPI function is disabled. 1: SPI function is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	SPPE	Parity Enable 0: A parity bit is not added to transmit data. Received-data parity check is not performed. 1: A parity bit is added to transmit data. Received-data parity check is performed.	R/W
9	SPOE	Parity Mode 0: Even parity is used for transmission and reception. 1: Odd parity is used for transmission and reception.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	PTE	Parity Self-Diagnosis Enable 0: Parity circuit self-diagnosis function is disabled. 1: Parity circuit self-diagnosis function is enabled.	R/W
12	SCKASE	RSPCK Auto-Stop Function Enable 0: RSPCK auto-stop function is disabled. 1: RSPCK auto-stop function is enabled.	R/W
13	BFDS	Between Burst Transfer Frames Delay Select 0: Delay (RSPCK delay, SSL negation delay and next-access delay) between frames is inserted in burst transfer 1: Delay between frames is not inserted in burst transfer.	R/W
14	MODFEN	Mode Fault Error Detection Enable 0: Mode fault error detection is disabled. 1: Mode fault error detection is enabled.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	SPEIE	SPI Error Interrupt Enable 0: SPI error interrupt request is disabled. 1: SPI error interrupt request is enabled.	R/W
17	SPRIE	SPI Receive Buffer Full Interrupt Enable 0: SPI receive buffer full interrupt request is disabled. 1: SPI receive buffer full interrupt request is enabled.	R/W
18	SPIIE	SPI Idle Interrupt Enable 0: Idle interrupt request is disabled. 1: Idle interrupt request is enabled.	R/W
19	SPDRES	SPI receive data ready error select Select the interrupt request to be generated when the reception data ready is detected 0: Receive data full interrupt 1: Error interrupt	R/W
20	SPTIE	SPI Transmit Buffer Empty Interrupt Enable 0: SPI transmit buffer empty interrupt request is disabled. 1: SPI transmit buffer empty interrupt request is enabled.	R/W
21	CENDIE	SPI Communication End Interrupt Enable 0: Communication end interrupt request is disabled. 1: Communication end interrupt request is enabled.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	SPMS	SPI Mode Select 0: SPI operation (4-wire) 1: Clock synchronous operation (3-wire)	R/W
25	SPFRF	SPI Frame Format Select 0: Motorola-SPI 1: TI-SSP Note: When SPMS = 1 (clock synchronous operation (3-wire)), this bit setting is invalid.	R/W

Bit	Symbol	Function	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
29:28	TXMD[1:0]	Communication Mode Select 0 0: Transmit-Receive 0 1: Transmit only Others: Receive only	R/W
30	MSTR	SPI Master/Slave Mode Select 0: Slave mode 1: Master mode	R/W
31	BPEN	Synchronization Circuit Bypass Enable 0: Non-Bypass 1: Bypass	R/W

Note: S-TYPE-3, P-TYPE-3

The SPI control register (SPCR) is used to set operating mode of the SPI. If the set BPEN, MSTR, TXMD[1:0], SPFRF, SPMS, MODFEN, BFDS, SCKASE, PTE, SPOE, SPPE bit value is modified while the SPE bit = 1, subsequent operation is not guaranteed.

### SPE bit (SPI Function Enable)

This bit is used to enable or disable SPI functions. Setting this bit to 1 enables SPI functions. When the MODF flag in the SPI status register (SPSR) is 1, the SPE bit is cleared to 0 and the SPE bit cannot be set to 1 until the MODF flag is cleared to 0. (See [section 34.3.10. Error Detection](#)) Setting the SPE bit to 0 disables SPI functions and initializes a part of module functions. (See [section 34.3.11. Initializing the SPI](#))

### SPPE bit (Parity Enable)

This bit is used to enable or disable the parity function.

### SPOE bit (Parity Mode)

This bit is used to specify even parity or odd parity.

In even parity mode, the parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an even number. In the same way, in odd parity mode, a parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an odd number. The SPOE bit is valid only when the SPPE bit in SPCR is set to 1.

### PTE bit (Parity Self-Diagnosis Enable)

This bit is used to enable or disable self-diagnosis of the parity circuit to confirm that the parity function is normal.

### SCKASE bit (RSPCK Auto-Stop Function Enable)

This bit is used to enable or disable the RSPCK auto-stop function. When this function is enabled, the RSPCK clock stops immediately before an overrun error occurs during data reception in master mode. For details, see [section 34.3.10.1. Overrun errors](#).

### BFDS bit (Between Burst Transfer Frames Delay Select)

This bit controls whether insert the delay time between the burst transfer frames.

Valid in the master mode (SPCR.MSTR = 1) for frames with the SPCMDn.SSLKP bit set to 1.

This bit should be set to 0 in slave mode. The usage of SSL delay control between transfer frames is shown as below. For details, see [section 34.3.12.1. Master mode operation](#).

1. Non-burst transmits
2. Burst transmit with delay between frames
  - 2-1. From the 1st frame to the last previous frame
  - 2-2. The last frame
3. Burst transmit with no delay between frames
  - 3-1. From the 1st frame to the last previous frame

## 3-2. The last frame

**Table 34.3 Usage of SSL delay control between transfer frames (Master mode)**

	SPCMDn. SSLKP bit	SPCR.BF DS bit	SSL delay control register*1 (RSPCK clock delay, SSL negation delay, next access delay)
1	0	0	Any given value. You can control each delay value according to setting for RSPCK clock delay, SSL negation delay and next access delay
2-1	1	0	
2-1	0	0	
3-1	1	1	Any given value. But delay is inserted only below. <ul style="list-style-type: none"> <li>RSPCK clock delay of the 1st frame</li> <li>SSL negation delay and next access delay of the last frame</li> </ul>
3-2	0	1	

Note 1. Whether the setting value of following bits are valid or not depends on the setting value of the SPCMD.SPNDEN bit. (See [section 34.2.6. SPCMDm : SPI Command Register \(m = 0 to 7\).](#))  
The SPDECR.SCKDL[2:0] bits: RSPCK delay  
The SPDECR.SLNDL[2:0] bits: SSL negate delay  
The SPDECR.SPNDL[2:0] bits: Next access delay

< Setting / operation example > (Motorola SPI, BFDS = 1 Case)

SPCMD0.SSLKP = 1 → Burst transfer / no interframe delay between 0 and 1 (SSL keep active)

SPCMD1.SSLKP = 1 → Burst transfer / no interframe delay between 1 and 2 (SSL keep active)

SPCMD2.SSLKP = 1 → Burst transfer / no interframe delay between 2 and 3 (SSL keep active)

SPCMD3.SSLKP = 1 → Burst transfer / no interframe delay between 3 and 4 (SSL keep active)

SPCMD4.SSLKP = 0 → do not Burst Transfer, and inactive SSL. (BFDS setting is invalid because it does not Burst Transfer.)

SPCMD5.SSLKP = 1 → Burst transfer / no interframe delay between 5 and 6 (SSL keep active)

SPCMD6.SSLKP = 1 → Burst transfer / no interframe delay between 6 and 7 (SSL keep active)

SPCMD7.SSLKP = 0 → do not Burst Transfer, and inactive SSL. (BFDS setting is invalid because it does not Burst Transfer.)

**MODFEN bit (Mode Fault Error Detection Enable)**

This bit is used to enable or disable detection of a mode fault error. (See [section 34.3.10. Error Detection.](#)) The SPI determines SSL0 pin input or output direction according to the combination of the MODFEN and MSTR bits. (See [section 34.3.2. Controlling the SPI Pins.](#))

**SPEIE bit (SPI Error Interrupt Enable)**

This bit is used to enable or disable an SPI error interrupt request when the SPI detects a mode fault error or an underrun error and sets the MODF flag in the SPI status register (SPSR) to 1, when the SPI detects an overrun error and sets the OVRF flag in SPSR to 1, or when the SPI detects a parity error and sets the PERF flag in SPSR to 1. (See [section 34.3.10. Error Detection](#))

**SPRIE bit (SPI Receive Buffer Full Interrupt Enable)**

This bit is used to enable or disable a receive buffer full interrupt request of the SPI.

**SPIIE bit (SPI Idle Interrupt Enable)**

This bit is used to enable or disable an idle interrupt request of the SPI after the SPI detects the idle state and sets the IDLNF flag in the SPI status register (SPSR) to 0.

**SPDRES bit (SPI receive data ready error select)**

When a receive data ready is detected (SPSR.SPDRF = 1), select whether to use SPI<sub>i</sub>\_SPRI (i = 0, 1) interrupt request or SPI<sub>i</sub>\_SPEI (i = 0, 1) interrupt request.

**SPTIE bit (SPI Transmit Buffer Empty Interrupt Enable)**

This bit is used to enable or disable a transmit buffer empty interrupt request of the SPI.



A transmit buffer empty interrupt request at the beginning of transmission is generated by setting the SPE bit to 1 simultaneously when or after the SPTIE bit is set to 1. Note that a transmit buffer empty interrupt is generated while the SPTIE bit is 1 even though SPI functions are disabled (SPE bit = 0).

#### CENDIE bit (SPI Communication End Interrupt Enable)

This bit controls generation of a communication end interrupt request.

#### SPMS bit (SPI Mode Select)

This bit is used to select SPI operation (4-wire) or clock synchronous operation (3-wire).

For clock synchronous operation, the SSL pin is not used but three pins RSPCK, MOSI, and MISO are used for communication. When SPMS = 1 (clock synchronous operation (3-wire)), the setting of the SPFRF bit is invalid.

To perform clock synchronous operation in master mode (SPCR.MSTR = 1), set the CPHA bit in the SPI command register (SPCMD) to 0 or 1. To perform clock synchronous operation in slave mode (SPCR.MSTR = 0), set the CPHA bit to 1. If this bit is set to 0 for clock synchronous operation in slave mode (SPCR.MSTR = 0), subsequent operation is not guaranteed.

The communication status according to the settings of the MSTR bit, TXMD[1:0] bits, SPFRF bit, and SPMS bit of the SPI control register (SPCR) as follows.

**Table 34.4 SPI Communication Status**

SPCR.MSTR	SPCR.TXMD[1]	SPCR.TXMD[0]	SPCR.SPFRF	SPCR.SPMS	Communication Status	Communication Status No
1	0	0	0	0	Transmit-Receive Master / Motorola SPI / SPI operation (4-wire)	1-(1)
1	0	0	1	0	Transmit-Receive Master / TI-SSP / SPI operation (4-wire)	1-(2)
1	0	0	—	1	Transmit-Receive Master / Clock synchronous operation (3-wire)	1-(3)
1	0	1	0	0	Transmit only Master / Motorola SPI / SPI operation (4-wire)	1-(4)
1	0	1	1	0	Transmit only Master / TI-SSP / SPI operation (4-wire)	1-(5)
1	0	1	—	1	Transmit only Master / Clock synchronous operation (3-wire)	1-(6)
1	1	—	0	0	Receive only Master / Motorola SPI / SPI operation (4-wire)	1-(7)
1	1	—	1	0	Receive only Master / TI-SSP / SPI operation (4-wire)	1-(8)
1	1	—	—	1	Receive only Master / Clock synchronous operation (3-wire)	1-(9)
0	0	0	0	0	Transmit-Receive Slave / Motorola SPI / SPI operation (4-wire) (default)	0-(1)
0	0	0	1	0	Transmit-Receive Slave / TI-SSP / SPI operation (4-wire)	0-(2)
0	0	0	—	1	Transmit-Receive Slave / Clock synchronous operation (3-wire)	0-(3)
0	0	1	0	0	Transmit only Slave / Motorola SPI / SPI operation (4-wire)	0-(4)
0	0	1	1	0	Transmit only Slave / TI-SSP / SPI operation (4-wire)	0-(5)
0	0	1	—	1	Transmit only Slave / Clock synchronous operation (3-wire)	0-(6)
0	1	—	0	0	Receive only Slave / Motorola SPI / SPI operation (4-wire)	0-(7)
0	1	—	1	0	Receive only Slave / TI-SSP / SPI operation (4-wire)	0-(8)
0	1	—	—	1	Receive only Slave / Clock synchronous operation (3-wire)	0-(9)

#### SPFRF bit (SPI Frame Format Select)

This bit selects the communication protocol.

The format of the SPI terminal (RSPCK, SSL0 to 7) can be set according to the set communication protocol.

When SPMS = 1 (clock synchronous operation (3-wire)), this bit is invalid because SSL is not used.

#### TXMD[1:0] bit (Communication Mode Select)

This bit is used to select the transmit-receive, transmit-only, and receive-only serial communication.

When TXMD[1:0] is set to 01 for communication, transmit-only is performed without reception.

When TXMD[1] is set to 1 for communication, receive-only is performed without transmission.

When TXMD[1:0] is set to 01 for communication, a receive buffer full interrupt request cannot be used.

When TXMD[1] is set to 1 for communication, a transmit buffer empty interrupt request cannot be used.

(See [section 34.3.6. Communication Operating Mode.](#))

**MSTR bit (SPI Master/Slave Mode Select)**

This bit is used to select master mode or slave mode of the SPI. The SPI determines input/output directions of pins RSPCK, MOSI, MISO, and SSL1 to SSL3 according to the MSTR bit setting.

**BPEN bit (Synchronization Circuit Bypass Enable)**

This bit selects whether to enable or disable the synchronization bypass function. Set this bit to 1 when the bus clock (PCLK) is also used as operation clock (TCLK), see [Figure 34.2.](#)

**34.2.4 SPCR2 : SPI Control Register 2**

Base address: SPln\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
 SPln\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPDRC[7:0]								RMST TG	RMED TG	—	RMFM[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	RMFM[4:0]	Frame processing count setting in Master Receive only The number of received frames can be adjusted in master receive only. 0x00: This function is not used*1 0x01: Automatically stop communication after processing 1 received frame ⋮ 0x1F: Automatically stop communication after processing 31 received frames	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	RMEDTG	End Trigger in Master Receive only 1: Receive End (Writable only when Master Receive only) Reading value is always 0	W
7	RMSTTG	Start Trigger in Master Receive only 1: Receive Start (Writable only when Master Receive only) Reading value is always 0	W
15:8	SPDRC[7:0]	SPI received data ready detect adjustment 0x00: Disable receive data ready detection function 0x01: Performs reception data ready judgment after 1 TCLK ⋮ 0xFF: Performs reception data ready judgment after 255 TCLK	R/W
16	SPLP	SPI Loopback 0: Normal mode 1: Loopback mode (inverted transmit data = receive data)	R/W
17	SPLP2	SPI Loopback 2 0: Normal mode 1: Loopback mode (transmit data = receive data)	R/W

Bit	Symbol	Function	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
20	MOIFV	MOSI Idle Fixed Value 0: The fixed value of MOSI idle = 0. 1: The fixed value of MOSI idle = 1.	R/W
21	MOIFE	MOSI Idle Fixed Value Enable 0: The MOSI output value is the last data of previous transfer. 1: The MOSI output value is the set MOIFV bit value.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. See SW flow in the [section 34.3.12.1. Master mode operation.](#)

#### RMFM[4:0] bit (Frame processing count setting in Master Receive only)

The number of received frames can be adjusted when operating in master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication operation mode select bits (SPCR.TXMD [1:0]) are 10b.

Only the start bit in master mode reception automatically stops communication after starts frame processing according to the value set in this bit after reception starts.

If the RMFM [4:0] bits are rewritten while the SPE bit of the SPI control register (SPCR) is 1, subsequent operations are not guaranteed.

#### RMEDTG bit (End Trigger in Master Receive only)

This bit is used to end reception when master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.TXMD [1:0]) are 10b.

#### RMSTTG bit (Start Trigger in Master Receive only)

This bit is used to start reception when master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.TXMD [1:0]) are 10b.

Writing 1 to this bit during reception is not accepted. Write again after reception is completed.

#### SPDRC[7:0] bit (SPI received data ready detect adjustment)

The receive data ready detection function can be disabled or, if used, the period until detection can be set from 1 to 255 TCLK.

The value set in the SPDRC [7:0] bits is used to 1 set the SPDRF flag. For details, see the description of SPDRF in [section 34.2.9. SPSR : SPI Status Register.](#)

If the set value is changed while the SPE bit is 1, subsequent operations are not guaranteed.

#### SPLP bit (SPI Loopback)

When the SPLP bit is set to 1, the SPI shuts down the route between the MISO pin and the shift register (when the MSTR bit in the SPI control register is 1) or shuts down the route between the MOSI pin and the shift register, inverts the input route value in the shift register, and then connects the route to the output route (when the MSTR bit in the SPI control register is 0) (loopback mode).

#### SPLP2 bit (SPI Loopback 2)

When the SPLP2 bit is set to 1, the SPI shuts down the route between the MISO pin and the shift register (when the MSTR bit in the SPI control register is 1) or shuts down the route between MOSI pin and the shift register and then connects the route to the output route without inverting the input route value in the shift register (when the MSTR bit in the SPI control register is 0) (loopback mode). If this bit is set to 1 together with the SPLP bit, setting this bit takes precedence.

#### MOIFV bit (MOSI Idle Fixed Value)

This bit is used to select the MOSI pin output value during the SSL negation period (including SSL retention period in burst transfer) when the MOIFE bit is 1 in master mode.

If this bit is modified with the SPE bit in the SPI control register (SPCR) set to 1, subsequent operation is not guaranteed.

**MOIFE bit (MOSI Idle Fixed Value Enable)**

This bit is used for the SPI in master mode to fix the MOSI output value during the SSL negation period (including SSL retention period in burst transfer). When MOIFE bit = 0, the SPI outputs the last data of the previous serial transfer to MOSI during the SSL negation period. When MOIFE bit = 1, the SPI outputs the fixed MOIFV bit value to MOSI.

If this bit is modified with the SPE bit in the SPI control register (SPCR) set to 1, subsequent operation is not guaranteed.

**34.2.5 SPCR3 : SPI Control Register 3**

Base address: SPIn\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
 SPIn\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SPSLN[2:0]		—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPBR[7:0]							—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P	
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSL0P	SSL0 Signal Polarity [In the Motorola-SPI case] 0: The SSL0 signal is active low. 1: The SSL0 signal is active high. [In the TI-SSP case] 0: The SSL0 signal is active high. 1: The SSL0 signal is active low.	R/W
1	SSL1P	SSL1 Signal Polarity [In the Motorola-SPI case] 0: The SSL1 signal is active low. 1: The SSL1 signal is active high. [In the TI-SSP case] 0: The SSL1 signal is active high. 1: The SSL1 signal is active low.	R/W
2	SSL2P	SSL2 Signal Polarity [In the Motorola-SPI case] 0: The SSL2 signal is active low. 1: The SSL2 signal is active high. [In the TI-SSP case] 0: The SSL2 signal is active high. 1: The SSL2 signal is active low.	R/W
3	SSL3P	SSL3 Signal Polarity [In the Motorola-SPI case] 0: The SSL3 signal is active low. 1: The SSL3 signal is active high. [In the TI-SSP case] 0: The SSL3 signal is active high. 1: The SSL3 signal is active low.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
15:8	SPBR[7:0]	SPI Bit Rate	R/W
23:16	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
26:24	SPSLN[2:0]	SPI Sequence Length 0 0 0: Sequence Length is 1 (Referenced SPCMDn, n = 0→0→...) 0 0 1: Sequence Length is 2 (Referenced SPCMDn, n = 0→1→0→...) 0 1 0: Sequence Length is 3 (Referenced SPCMDn, n = 0→1→2→0→...) 0 1 1: Sequence Length is 4 (Referenced SPCMDn, n = 0→1→2→3→0→...) 1 0 0: Sequence Length is 5 (Referenced SPCMDn, n = 0→1→2→3→4→0→...) 1 0 1: Sequence Length is 6 (Referenced SPCMDn, n = 0→1→2→3→4→5→0→...) 1 1 0: Sequence Length is 7 (Referenced SPCMDn, n = 0→1→2→3→4→5→6→0→...) 1 1 1: Sequence Length is 8 (Referenced SPCMDn, n = 0→1→2→3→4→5→6→7→0→...)	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

### SSLiP bits (SSL Signal Polarity Bits)

These bits are used to specify the polarity of SSL signals. The set SSLiP bit (i = 0 to 3) values indicate the active polarity of SSLi signals.

If any of these SSLiP bits is modified with the SPE bit in the SPI control register (SPCR) set to 1, subsequent operation is not guaranteed.

Note: SSL0 is different from SSL1-SSL3. When slave or multi-master, it functions as an input.

For details, see [section 34.3.3.2. Single-master/single-slave with the MCU as a slave](#), and [section 34.3.3.5. Multi-master/multi-slave with the MCU as a master](#).

### SPBR[7:0] bit (SPI Bit Rate)

The SPI bit rate bits (SPBR) is used to set the bit rate in master mode. If SPBR is modified while the MSTR bit in the SPI control register (SPCR) is 1, subsequent operation is not guaranteed.

When the SPI is used in slave mode, the bit rate depends on the input clock bit rate regardless of the SPCMD.BRDV setting. (Specify a bit rate that meets electrical characteristics.)

The bit rate is determined by a combination of the set SPBR value and the set BRDV[1:0] bits value in the SPI command register (SPCMD0 to SPCMD7).

The bit rate is calculated by the following expression, where n is the set SPBR value (0 to 255) and N is the set BRDV[1:0] bits value (0 to 3).

$$\text{Bit rate} = \frac{f(\text{TCLK})}{2 \times (n + 1) \times 2^N}$$

The following table shows an example of correspondence between bit rates and set values of SPBR and BRDV[1:0].

**Table 34.5 Corresponding Between Bit Rates and Set Values (Example)**

SPBR Value (n)	BRDV Value (N)	Division Ratio	Bit Rate				
			TCLK = 32 MHz	TCLK = 36 MHz	TCLK = 40 MHz	TCLK = 50 MHz	TCLK = 120MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps	60.0Mbps
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps	30.0Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps	20.0Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps	15.0Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps	12.0Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps	10.0Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps	5.0Mbps
5	2	48	677 kbps	750 kbps	833 kbps	1.04 Mbps	2.5Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps	1.25Mbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps	29.3kbps

### SPSLN[2:0] bit (SPI Sequence Length)

These bits are used to set the sequence length for the SPI in master mode to perform sequence operation. According to the sequence length specified by SPSLN[2:0] bits, the SPI in master mode changes SPI command registers 0 to 7 (SPCMD0 to SPCMD7) to be referenced and the reference sequence. For details, see [section 34.3.13.1. Master mode operation](#).

The SPI in slave mode always references SPCMD0.

### 34.2.6 SPCMDm : SPI Command Register (m = 0 to 7)

Base address: SPln\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
 SPln\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x14 + 0x04 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	SSLA[2:0]			—	—	—	SPB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCKD EN	SLND EN	SPND EN	LSBF	—	—	—	—	SSLK P	—	—	—	BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPHA	RSPCK Phase 0: Data is sampled at an odd edge and changes at an even edge. 1: Data changes at an odd edge and is sampled at an even edge.	R/W
1	CPOL	RSPCK Polarity 0: RSPCK in idle state is 0. 1: RSPCK in idle state is 1.	R/W
3:2	BRDV[1:0]	Bit Rate Division 0 0: Base bit rate 0 1: Base bit rate divided by 2 1 0: Base bit rate divided by 4 1 1: Base bit rate divided by 8	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	SSLKP	SSL Signal Level Hold 0: All SSL signals are negated at the end of transfer. 1: SSL signal level is held after the transfer ends until the next access starts.	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	LSBF	SPI LSB First 0: MSB first 1: LSB first	R/W
13	SPNDEN	SPI Next-Access Delay Enable 0: Next-access delay is 1RSPCK + 5TCLK 1: Next-access delay is the set value of the SPI next-access delay (SPDECR.SPNDL).	R/W
14	SLNDEN	SSL Negation Delay Setting Enable 0: [Master] SSL negation delay is 1RSPCK. [Slave in the TI-SSP] SSL negation delay is 1TCLK 1: SSL negation delay is the set value of the slave select negation delay (SPDECR.SLNDL).	R/W

Bit	Symbol	Function	R/W
15	SCKDEN	RSPCK Delay Setting Enable [In the Motorola-SPI case] 0: RSPCK delay is 1 RSPCK. 1: RSPCK delay is the set value of the RSPCK delay (SPDECR.SCKDL). [In the TI-SSP case] 0: RSPCK delay is 0 RSPCK. 1: RSPCK delay is the set value of the RSPCK delay (SPDECR.SCKDL).	R/W
20:16	SPB[4:0]	SPI Data Length 0x00 to 0x02: Setting prohibited 0x03: 4bits 0x04: 5bits 0x05: 6bits ⋮ 0x1E: 31bits 0x1F: 32bits	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
26:24	SSLA[2:0]	SSL Signal Assertion 0 0 0: SSL0 0 0 1: SSL1 0 1 0: SSL2 0 1 1: SSL3 Others: Setting prohibited	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SPI has eight SPI command registers (SPCMD0 to SPCMD7) that are used to set the transfer format of the SPI in master mode. Furthermore, some bits in SPCMD0 are used to set the transfer format of the SPI in slave mode. The SPI in master mode sequentially references SPCMD0 to SPCMD7 according to the setting of the SPSLN[2:0] bits in the SPI Control register 3 (SPCR3), and then performs serial transfer specified in the referenced SPCMD.

SPI set the SPCMD register before setting data to be transmitted by referencing the SPCMD when the transmit buffer is empty (while the next-transfer data has not been set).

The SPCMD referenced by the SPI in master mode is indicated by SPCP[2:0] in the SPI status register (SPSR). If SPCMD0 is modified while the SPI in slave mode is enabled (SPCR.SPE = 1), subsequent operation is not guaranteed.

#### CPHA bit (RSPCK Phase)

This bit is used to set the RSPCK phase of the SPI in master mode or slave mode. To perform data communication between SPI modules, the same RSPCK phase must be set for both modules.

When SPCR.SPMS = 0 and SPCR.SPFRF = 1 (in TI SSP mode), setting CPHA = 0 is invalid.

#### CPOL bit (RSPCK Polarity)

This bit is used to set the RSPCK polarity of the SPI in master mode or slave mode. To perform data communication between SPI modules, the same RSPCK polarity must be set for both modules.

#### BRDV[1:0] bit (Bit Rate Division)

This register is used to determine the bit rate with a combination of the set values of the BRDV[1:0] bits and the SPI bit rate register (SPCR3.SPBR). The set SPBR value determines the base bit rate. The set BRDV[1:0] bits value is used to select undivided, 2-divided, 4-divided, or 8-divided base bit rate. SPCMD0 to SPCMD7 enable setting of different BRDV[1:0] values. This makes it possible to perform serial transfer with a different bit rate for each command.

#### SSLKP bit (SSL Signal Level Hold)

This bit is used to set whether to hold or negate the SSL signal level of the current command during a period from the SSL negation timing for the current command to the SSL assertion timing for the next command when the SPI in master mode performs serial transfer. Setting this bit to 1 enables burst transfer in SPI operation master mode. For details, see [section 34.3.12.1. Master mode operation](#).

To use the SPI in slave mode, set SSLKP bit to 0.

**LSBF bit (SPI LSB First)**

This bit is used to set the data format of the SPI in master mode or slave mode to MSB first or LSB first.

**SPNDEN bit (SPI Next-Access Delay Enable)**

This bit is used to set the period (next-access delay) after the SPI in master mode inactivates the SSL signal at the end of serial transfer until it enables SSL signal assertion of the next access. When SPNDEN bit = 0, the SPI sets the next-access delay to 1 RSPCK + 5TCLK. When SPNDEN bit = 1, the SPI inserts the next-access delay in accordance with the SPI next-access delay register (SPDECR.SPNDL) setting.

To use the SPI in slave mode, set SPNDEN bit to 0.

**SLNDEN bit (SSL Negation Delay Setting Enable)**

[In the Motorola-SPI case]

This bit is used to set the period (SSL negation delay) after the SPI in master mode stops RSPCK oscillation until it inactivates the SSL signal. When SLNDEN bit = 0, the SPI sets the SSL negation delay to 1 RSPCK. When SLNDEN bit = 1, the SPI negates the SSL signal with the RSPCK delay in accordance with the slave select negation delay register (SPDECR.SLNDL) setting.

To use the SPI in slave mode, set SLNDEN bit to 0.

[In the TI-SSP case]

This bit is used to set the period from when the master mode SPI stops RSPCK oscillation to when the OE signal is inactivated, or when the slave mode SPI detects the last edge of RSPCK and then negates the OE signal. When the SLNDEN bit is 0, the SSL negate delay is 1 RSPCK in master mode and 1 TCLK in slave mode. When SLNDEN bit = 1, the SPI negates the SSL signal with the RSPCK delay in accordance with the slave select negation delay register (SPDECR.SLNDL) setting.

When using SPI in slave mode except TI SSP setting, set the SLNDEN bit to 0.

**SCKDEN bit (RSPCK Delay Setting Enable)**

[In the Motorola-SPI case]

This bit is used to set the period (RSPCK delay) after the SPI in master mode activates the SSL signal until it oscillates RSPCK. When SCKDEN bit = 0, the SPI sets the RSPCK delay to 1 RSPCK. When SCKDEN bit = 1, the SPI starts RSPCK oscillation with the RSPCK delay in accordance with the RSPCK delay register (SPDECR.SCKDL) setting.

To use the SPI in slave mode, set SCKDEN bit to 0.

[In the TI-SSP case]

This bit is used to set the period from the start of assertion of the SSL signal to the RSPCK oscillation (RSPCK delay) and the period of the SSL signal to negation by the SPI in master mode. When SCKDEN bit = 0, the SPI does not set the RSPCK delay. When SCKDEN bit = 1, the SPI starts RSPCK oscillation with the RSPCK delay in accordance with the RSPCK delay register (SPDECR.SPCKDL) setting.

To use the SPI in slave mode, set SCKDEN bit to 0.

**SPB[4:0] bit (SPI Data Length)**

These bits are used to set the transfer data length of the SPI in master mode or slave mode.

**SSLA[2:0] bit (SSL Signal Assertion)**

These bits are used to control SSL signal assertion for the SPI in master mode to perform serial transfer. The set SSLA[2:0] bits value controls assertion of the SSL3 to SSL0 signals. The signal polarity when the SSL signal is asserted depends on the set value of the SPI slave select polarity register (SPCR3.SSLiP). When SSLA[2:0] bits are set to 000b in multi-master mode, serial transfer is performed with all SSL signals negated (because SSL0 is input).

To use the SPI in slave mode, set SSLA[2:0] bits to 000b.



### 34.2.7 SPDCR : SPI Data Control Register

Base address: SPIn\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
 SPIn\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SPFC[1:0]	—	—	—	SINV	SPRD TD	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BYSW	Byte Swap Operating Mode Select 0: Byte Swap OFF 1: Byte Swap ON	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	SPRDTD	SPI Receive Data or Transmit Data Select 0: The SPDR reads the receive buffer. 1: The SPDR reads the transmit buffer	R/W
4	SINV	Serial data invert bit 0: Not invert serial data 1: Invert serial data.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SPFC[1:0]	Frame Count 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The SPI data control register (SPDCR) controls the data format.

If the value set in this register is changed while the SPE bit is 1, subsequent operations are not guaranteed.

#### **BYSW bit (Byte Swap Operating Mode Select)**

It is a setting bit, that is to swap a transmit/receive data in byte units. A data after byte swap is different by a data length (setting of SPCMD.SPB[4:0]).

When byte swap, A data length (setting of SPB[4:0]) must be set to 32bit or 16bit. Other case of data length (i.e. 4 to 15, 17 to 31-bit length), byte swap is not guaranteed. For the arrangement of data before and after swapping data lengths of 32 bits and 16 bits, see [section 34.3.4.3. Byte Swap Transmission](#) and [section 34.3.4.4. Byte Swap Reception](#).

When the parity function set to valid, the behavior is not guaranteed.

#### **SPRDTD bit (SPI Receive Data or Transmit Data Select)**

This bit is used to select receive buffer or transmit buffer from which the SPI data register (SPDR) value is read.

When the transmit buffer is read, the value that was written to SPDR immediately before is read.

#### **SINV bit (Serial data invert bit)**

This bit is used to invert transmit data and receive data.

When the SINV bit is set to 1, transmit buffer (SPTX) data is inverted to invert transmit data and receive data, and then the inverted data is stored in the receive buffer (SPRX). The parity bit is the value corresponding to the inverted transmission/reception data.

### SPFC[1:0] bit (Frame Count)

Used for the condition to set the CENDF flag in slave receive only mode.

For details on the CENDF flag setting conditions, see [section 34.2.9. SPSR : SPI Status Register](#).

Note that this bit is invalid except in the slave receive only mode.

## 34.2.8 SPDCR2 : SPI Data Control Register 2

Base address: SPIn\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
SPIn\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x44

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TTRG[1:0]		—	—	—	—	—	—	RTRG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RTRG[1:0]	Receive FIFO threshold setting 0 0: threshold 0 0 1: threshold 1 1 0: threshold 2 1 1: threshold 3	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	TTRG[1:0]	Transmission FIFO threshold setting 0 0: threshold 0 0 1: threshold 1 1 0: threshold 2 1 1: threshold 3	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

SPI data control register 2 (SPDCR2) controls the FIFO threshold. If the value set in this register is changed while the SPE bit is 1, subsequent operations are not guaranteed.

### RTRG[1:0] bit (Receive FIFO threshold setting)

Set the receive FIFO threshold.

When the number of data stored in the receive FIFO > the number of frames set by RTRG[1:0], the receive buffer full flag is set.

### TTRG[1:0] bit (Transmission FIFO threshold setting)

Set the transmit FIFO threshold.

When the number of empty stages in the transmit FIFO > the number of frames set in TTRG[1:0], the transmit buffer empty flag is set.

### 34.2.9 SPSR : SPI Status Register

Base address: SPI<sub>n</sub>\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
 SPI<sub>n</sub>\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SPRF	CEND F	SPTE F	UDRF	PERF	MODF	IDLNF	OVRF	SPDR F	—	—	—	—	—	—	—
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SPECM[2:0]			—	SPCP[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R
10:8	SPCP[2:0]	SPI Command Pointer 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
11	—	This bit is read as 0. The write value should be 0.	R
14:12	SPECM[2:0]	SPI Error Command 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
22:15	—	These bits are read as 0. The write value should be 0.	R
23	SPDRF	SPI Receive Data Ready Flag 0: Receive data ready not detected 1: Receive data ready detected	R
24	OVRF	Overrun Error Flag 0: No overrun error is present. 1: An overrun error is present.	R
25	IDLNF	SPI Idle Flag 0: The SPI is in the idle state. 1: The SPI is in the transfer state.	R
26	MODF	Mode Fault Error Flag 0: Neither mode fault error nor underrun error is present. 1: A mode fault error or underrun error is present.	R
27	PERF	Parity Error Flag 0: No parity error is present. 1: A parity error is present.	R
28	UDRF	Underrun Error Flag 0: When MODF=0, neither mode fault error nor underrun error is present. When MODF=1, a mode fault error is present. 1: When MODF=0, neither mode fault error nor underrun error is present. When MODF=1, an underrun error is present.	R

Bit	Symbol	Function	R/W
29	SPTEF	SPI Transmit Buffer Empty Flag 0: The number of empty stages in the transmit FIFO $\leq$ the value set in SPDCR2.TTRG 1: The number of empty stages in the transmit FIFO $>$ the value set in SPDCR2.TTRG	R
30	CENDF	Communication End Flag 0: The SPI is not communicating or communicating. 1: The SPI communication completed.	R
31	SPRF	SPI Receive Buffer Full Flag 0: The number of data stored in the receive FIFO $\leq$ number of frames set by the SPDCR2.RTRG bit. 1: The number of data stored in the receive FIFO $>$ number of frames set by the SPDCR2.RTRG bit.	R

Note: S-TYPE-3, P-TYPE-3

The SPI status register (SPSR) stores flags that indicate SPI's operating status.

### SPCP[2:0] bit (SPI Command Pointer)

These bits indicate SPI command registers 0 to 7 (SPCMD0 to SPCMD7) indicated by the current pointer in the SPI sequence control. For details about the SPI sequence control, see [section 34.3.13.1. Master mode operation](#).

### SPECM[2:0] bit (SPI Error Command)

These bits indicate SPI command registers 0 to 7 (SPCMD0 to SPCMD7) indicated by the command pointer (SPCP[2:0] bits) when an error was detected in the SPI sequence control. The SPI updates the SPECM[2:0] bits value only when an error is detected. When no error is present (OVRF, MODF, and PERF flags in SPSR are 0), the SPECM[2:0] bits value has no meaning. For the SPI's error detection function, see [section 34.3.10. Error Detection](#). For the SPI's sequence control, see [section 34.3.13.1. Master mode operation](#).

### SPDRF bit (SPI Receive Data Ready Flag)

During communication (SPCR.SPE = 1), a certain period of time has elapsed while the number of data stored in the reception FIFO  $\leq$  the reception FIFO threshold.

This bit is set to 0 when the reception operation is not performed (SPCR.TXMD[1:0] = 01b).

[Setting condition]

All the following two conditions are met.

- SPCR2.SPDRC[7:0]  $\neq$  0x00.
- After the receive FIFO has been written, when the number of data stored in the receive FIFO  $\leq$  the receive FIFO threshold and the value set by SPDRC[7:0] has elapsed

[Clearing condition]

- When 1 is written to the SPSRC.SPDRFC bit.

### OVRF bit (Overrun Error Flag)

This flag indicates whether an overrun error is present. When the RSPCK clock auto-stop function is enabled (SPCR.SCKASE = 1) in master mode (SPCR.MSTR = 1), no overrun error occurs and, therefore, this flag is not set to 1. For details, see [section 34.3.10.1. Overrun errors](#).

[Setting condition]

When serial transfer is completed in one of the following two conditions with data stored in the receive FIFO for the number of FIFO stages.

- SPCR.TXMD[1:0] = 00b. (transmit-receive mode)
- SPCR.TXMD[1:0] = 10b. (receive only)

[Clearing condition]

- When 1 is written to the SPSRC.OVRFC bit.

**IDLNF bit (SPI Idle Flag)**

This flag indicates transfer status of the SPI.

[Setting condition]

[Transmit-Receive, Transmit-only Master mode]

- None of the clearing conditions (Transmit-Receive / Transmit-only in Master mode) below is met.

[Receive-only Master mode]

- When 1 is written to RMSTTG of SPCR2.

[Slave mode]

- The SPE bit in SPCR is 1 (SPI function enabled).

[Clearing conditions]

Communication status: 1-(1) to (6) \* For details of communication status, see [Table 34.4](#).

[Transmit-Receive, Transmit-only Master mode]

Any of the following two conditions is met.

- The SPE bit in SPCR is 0 (SPI initialization).
- All the following three conditions are met.
  - The next transfer data is not set in the transmission buffer (SPTXn, n = 0 to 3)
  - The SPCP bits in SPSR are 000b (at the beginning of sequence control).
  - The operation completed by the next access delay (the master main state machine has transitioned to the idle state)

[Receive-only Master mode]

Communication status: 1-(7) to (9)

Any of the following two conditions is met.

- The SPE bit in SPCR is 0 (SPI initialization).
- Any of the following 3 conditions is met.
  - When RMFM[4:0] = 0x00, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
  - When RMFM[4:0] ≠ 0x00, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
  - When RMFM[4:0] ≠ 0x00, the operation completed by the next access delay after processing is completed for the number of received frames set in RMFM[4:0] (the master main state machine has transitioned to the idle state)

[Slave mode]

Communication status: 0-(1) to (9)

- The SPE bit in SPCR is 0 (SPI initialization).

**MODF bit (Mode Fault Error Flag)**

This flag indicates whether a mode fault error or an underrun error is present. The UDRF flag allows you to see which error (mode fault error or underrun error) has occurred.

[Setting condition]

[Multi-master mode]

- The SSL0 pin input level becomes active level while the SPCR.MSTR bit = 1 (master mode) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.

[Slave, Motorola-SPI mode]

Any of the following two conditions is met.

- The SSL0 pin is negated before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0 (slave mode), SPCR.SPFRF bit = 0 (Motorola-SPI) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

[Slave, TI-SSP mode]

Any of the following two conditions is met.

- The SSL0 pin is asserted before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0 (slave mode), SPCR.SPFRF bit = 1 (TI-SSP) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

The SSL0 signal active level depends on the SPCR3.SSLiP bits (SSL signal polarity bits).

[Clearing condition]

- When 1 is written to the SPSRC.MODFC bit.

### PERF bit (Parity Error Flag)

This flag indicates whether a parity error is present.

[Setting condition]

When the serial transfer ends and a parity error is detected with the SPPE bit of SPCR set to 1 under any of the following 2 conditions.

- SPCR.TXMD[1:0] = 00b. (transmit-receive master mode or transmit-receive slave mode)
- SPCR.TXMD[1:0] = 10b. (receive-only master mode or receive-only slave mode)

[Clearing condition]

- When 1 is written to the SPSRC.PERFC bit.

### UDRF bit (Underrun Error Flag)

This flag indicates that a mode fault error or an underrun error is present.

[Setting condition]

- Serial transfer is started before transmit data output becomes ready while the SPCR.MSTR bit = 0 and the SPCR.TXMD[1:0] bit = 00b or 01b (transmit-receive slave mode or transmit-only slave mode) and the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

[Clearing condition]

- When 1 is written to the SPSRC.UDRFC bit.

### SPTEF bit (SPI Transmit Buffer Empty Flag)

This flag indicates the transmit buffer (SPTX) status in the SPI data register (SPDR).

[Setting condition]

Any of the following 3 conditions is met.

- The SPE bit is set to 0 (SPI initialization).
- When the number of empty transmission FIFO stages > the value set in SPDCR2.TTRG[1:0].
- When 1 is written to SPFCR.SPFRST.

[Clearing condition]

Any of the following two conditions is met.

- At the time of final access when transmission data is written to SPDR (SPTXn, n = 0 to 3) in one processing routine using DTC / DMAC.
- When 1 is written to the SPSRC.SPTEFC bit.

Writing a value to the SPDR register is enabled only while the SPTEF flag = 1. If a value is written to the SPDR register while the SPTEF flag = 0, transmit buffer data is not updated.

### CENDF bit (Communication End Flag)

This flag indicates communication end status of SPI. It turns 1 at communication end and turns 0 at starting next communication.

[Setting condition]

[Transmit-Receive / Transmit-only Master mode]

Communication status: 1-(1) to (6) \* For details of communication status, see [Table 34.4](#).

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer (SPTXn, n = 0 to 3)
- The SPSR.SPCP[2:0] are 000b. (It means the head of the sequential control.)
- The operation completed by the next access delay (the master main state machine has transitioned to the idle state)

[Receive-only Master mode]

Communication status: 1-(7) to (9)

Any of the following 3 conditions is met.

- When RMFM[4:0] = 0x00, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
- When RMFM[4:0] ≠ 0x00, after writing 1 to RMEDTG, the operation completed by the next access delay (the master main state machine has transitioned to the idle state)
- When RMFM[4:0] ≠ 0x00, the operation completed by the next access delay after processing is completed for the number of received frames set in RMFM[4:0] (the master main state machine has transitioned to the idle state)

[Transmit-Receive / transmit-only slave, Motorola-SPI mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(1), (4)

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty. (It means SPI does not do serial transfer.)
- SSL0 was negated.

[Transmit-Receive / transmit-only slave, TI-SSP mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(2), (5)

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty. (It means SPI does not do serial transfer.)
- When the SSL0 negate delay is completed.

[Transmit-Receive / transmit only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1)]

Communication status: 0-(3), (6)

The following 3 conditions are met.

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty. (It means SPI does not do serial transfer.)
- The last even edge of RSPCK of the frame was detected. (When the SPCMD.CPHA bit is 1.)

[Receive only slave, Motorola-SPI mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(7)

The following condition is met.

- SSL0 input was negated after getting frames for SPDCR.SPFC set value in the receive buffer.

[Receive only slave, TI-SSP mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(8)

The following condition is met.

- SSL0 negate delay is completed after getting frames for SPDCR.SPFC set value in the receive buffer.

[Receive only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1)]

Communication status: 0-(9)

The following condition is met.

- The last even edge of RSPCK of the Last frame received for SPFC sets value. (When the SPCMD.CPHA bit is 1.)

[Clearing condition]

[Transmit-Receive / Transmit-only Master mode]

Communication status: 1-(1) to (6)

Any of the following 2 conditions is met.

- The next transmit data was written to the transmit buffer (SPTX).
- When 1 is written to the SPSRC.CENDFC bit.

[Receive -only Master mode]

Communication status: 1-(7) to (9)

Any of the following 2 conditions is met.

- When 1 is written to the SPCR2.RMSTTG bit with SPE = 1.
- When 1 is written to the SPSRC.CENDFC bit.

[Transmit-receive / transmit only slave mode]

Communication status: 0-(1) to (6)

Satisfy one of following.

- The next transmit data was written to the transmit buffer (SPTX).
- When 1 is written to the SPSRC.CENDFC bit.

[Receive only slave mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0)]

Communication status: 0-(7) to (8)

Satisfy one of following.

- SSL0 assertion of next data was detected.
- When 1 is written to the SPSRC.CENDFC bit.

[Receive only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1)]

Communication status: 0-(9)

Satisfy one of following.

- The first edge of RSPCK of the next data was detected.
- When 1 is written to the SPSRC.CENDFC bit.



### SPRF bit (SPI Receive Buffer Full Flag)

This flag indicates the receive buffer (SPRX) status in the SPI data register (SPDR).

[Setting condition]

When the number of data stored in the receive FIFO > the number of frames set in the SPDCR2.RTRG[1:0] bits in Transmit-Receive, receive-only mode. However, the SPRF flag does not change from 0 to 1 while the OVRF flag = 1. (See section 34.3.10. Error Detection.)

[Clearing condition]

Any of the following 3 conditions is met.

- At the last access when read data is read from SPDR (SPRXn, n = 0 to 3) in one processing routine using DTC / DMAC
- When 1 is written to the SPSRC.SPRFC bit
- When 1 is written to the SPFCR.SPFRST bit

### 34.2.10 SPTFSR : SPI Transfer FIFO Status Register

Base address: SPIn\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
 SPIn\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x58

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TFDN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TFDN[2:0]	Transmit FIFO data empty stage number 0 0 0: Number of empty stages 0 ⋮ 1 0 0: Number of empty stages 4	R
31:3	—	These bits are read as 0. The write value should be 0.	R

### TFDN[2:0] bit (Transmit FIFO data empty stage number)

Displays the number of empty transmission FIFO stages. By clearing the SPCR.SPE bit, TFDN[2:0] will be the initial value after reset (= all empty).

### 34.2.11 SPRFSR : SPI Receive FIFO Status Register

Base address: SPIn\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
 SPIn\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x5C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	RFDN[2:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	RFDN[2:0]	Receive FIFO data store stage number 0 0 0: Number of store stages 0 ⋮ 1 0 0: Number of store stages 4	R
31:3	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3

**RFDN[2:0] bit (Receive FIFO data store stage number)**

Displays the number of stores receive FIFO stages. RFDN [2:0] is cleared by clearing the SPCR.SPE bit.

**34.2.12 SPPSR : SPI Polling Register**

Base address: SPIn\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
SPIn\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x60

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEP S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPEPS	SPI Polling Status 0: SPCR.SPE is 0 1: SPCR.SPE is 1	R
31:1	—	These bits are read as 0. The write value should be 0.	R

**SPEPS bit (SPI Polling Status)**

This bit indicates status of SPCR.SPE bit after synchronization from bus clock (PCLK) to operation clock (TCLK).

**34.2.13 SPSRC : SPI Status Clear Register**

Base address: SPIn\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
SPIn\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SPRF C	CEND FC	SPT FC	UDRF C	PERF C	MODF C	—	OVRF C	SPDR FC	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
22:0	—	The write value should be 0.	W
23	SPDRFC	SPI Receive Data Ready Flag Clear By writing 1, the SPI Receive Data Ready Flag can be cleared. Reading value is always 0.	W

Bit	Symbol	Function	R/W
24	OVRFC	Overrun Error Flag Clear By writing 1, the Overrun Error Flag can be cleared. Reading value is always 0.	W
25	—	The write value should be 0.	W
26	MODFC	Mode Fault Error Flag Clear By writing 1, the Mode Fault Error Flag can be cleared. Reading value is always 0.	W*1
27	PERFC	Parity Error Flag Clear By writing 1, the Parity Error Flag can be cleared. Reading value is always 0.	W
28	UDRFC	Underrun Error Flag Clear By writing 1, the Underrun Error Flag can be cleared. Reading value is always 0.	W*2
29	SPTEFC	SPI Transmit Buffer Empty Flag Clear By writing 1, the SPI Transmit Buffer Empty Flag can be cleared. Reading value is always 0.	W
30	CENDFC	Communication End Flag Clear By writing 1, the Communication End Flag can be cleared. Reading value is always 0.	W
31	SPRFC	SPI Receive Buffer Full Flag Clear By writing 1, the SPI Receive Buffer Full Flag can be cleared. Reading value is always 0.	W

Note: S-TYPE-3, P-TYPE-3

Note 1. Before setting MODFC and UDRFC, make sure that SPSR.MODF and UDRF are set to 1.

Note 2. When clearing the UDRF flag, clear the MODF flag at the same time (MODFC = 1).

The SPI status clear register (SPSRC) is a register that clears the status flag (SPSR) that indicates the operating status of SPI.

### 34.2.14 SPFCR : SPI FIFO Clear Register

Base address: SPI<sub>n</sub>\_B = 0x4035\_C000 + 0x0100 × n (n = 0, 1)  
SPI<sub>n</sub>\_B\_NS = 0x5035\_C000 + 0x0100 × n (n = 0, 1)

Offset address: 0x6C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPFR ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPFRST	SPI FIFO clear By writing 1, the pointer in the FIFO and the stored data are initialized. Reading value is always 0.	W
31:1	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

The FIFO clear register (SPFCR) is used to clear the FIFO.

If SPFCR is rewritten while the SPE bit of the SPI control register (SPCR) is 1, subsequent operations are not guaranteed.

#### SPFRST bit (SPI FIFO clear)

Initializing the pointer and stored data in the transmit / receive FIFO by writing 1

## 34.3 Operation

In this section, the serial transfer period refers to the period from the beginning of driving valid data to the fetching of the final valid data.

### 34.3.1 Overview of SPI Operation

The SPI is capable of synchronous serial transfers in the following modes:

- Slave mode (SPI operation)
- Single-master mode (SPI operation)
- Multi-master mode (SPI operation)
- Slave mode (clock synchronous operation)
- Master mode (clock synchronous operation)

The SPI mode can be selected by using the MSTR, MODFEN, SPMS, and SPFRF bits in SPCR. Table 34.6 lists the relationship between SPI modes and SPCR settings, and a description of each mode.

**Table 34.6 Relationship between SPCR settings and SPI modes (1 of 2)**

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
SPFRF bit setting	valid	valid	valid	Invalid	Invalid
RSPCKn pins	Input	Output	Output/Hi-Z	Input	Output
MOSIn pin	Input	Output	Output/Hi-Z	Input	Output
MISOOn pin	Output/Hi-Z	Input	Input	Output	Input
SSLn0 pins	Input	Output	Input	Hi-Z <sup>*1</sup>	Hi-Z <sup>*1</sup>
SSLn1 to SSLn3 pins	Hi-Z <sup>*1</sup>	Output	Output/Hi-Z	Hi-Z <sup>*1</sup>	Hi-Z <sup>*1</sup>
SSL polarity change function	Supported	Supported	Supported	—	—
Max transfer rate	TCLK/2	TCLK/2	TCLK/2	TCLK/2	TCLK/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two				
Clock phase	Two <sup>*6</sup>	Two <sup>*6</sup>	Two <sup>*6</sup>	One (CPHA = 1)	Two
Transfer data length	4 to 32 bits				
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0, 1)	Possible (CPHA = 0, 1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported <sup>*7</sup>	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer trigger	SSL input active or RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)	RSPCK oscillation	Write to transmit buffer on generation of transmit buffer empty interrupt request (SPTEF = 1)
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported <sup>*5</sup>				
Receive buffer full detection	Supported <sup>*2</sup>				
Overrun error detection	Supported <sup>*2</sup>	Supported <sup>*2 *4</sup>	Supported <sup>*2 *4</sup>	Supported <sup>*2</sup>	Supported <sup>*2</sup>

**Table 34.6 Relationship between SPCR settings and SPI modes (2 of 2)**

Mode	Slave (SPI operation)	Single-master (SPI operation)	Multi-master (SPI operation)	Slave (clock synchronous operation)	Master (clock synchronous operation)
Parity error detection	Supported*3 *2				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported*5	Not supported	Not supported	Supported*5	Not supported

Note 1. This function is not supported in this mode.

Note 2. When SPI is transmit-master mode or transmit-slave mode (see Table 34.4), detection of receive buffer full, overrun error, and parity error are not performed.

Note 3. When the SPCR.SPPE bit is 0, parity error detection is not performed.

Note 4. When the SPCR.SCKASE bit is 1, overrun error detection does not proceed.

Note 5. When SPI is receive only slave mode, none of transmit buffer empty and underrun error is detected.

Note 6. CPHA = 0 is invalid in TI SSP mode. (If it is set, the operation is the same as when CPHA = 1.)

Note 7. Supported only in TI SSP mode.

### 34.3.2 Controlling the SPI Pins

Based on the settings of the MSTR, MODFEN, and SPMS bits in SPCR and the PmnPFS.NCODR bit for I/O Ports, the SPI can switch pin states. Table 34.7 lists the relationship between pin states and bit settings. Setting the PmnPFS.NCODR bit for an I/O port to 0 selects the CMOS output. Setting it to 1 selects the open-drain output. The I/O port settings must follow this relationship.

**Table 34.7 Relationship between pin states and bit settings (1 of 2)**

Mode	Pin	Pin state*2	
		PmnPFS.NCODR bit for I/O ports = 0	PmnPFS.NCODR bit for I/O ports = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3	CMOS output	Open-drain output
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLn0	Input	Input
	SSLn1 to SSLn3*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISOn	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKn	Input	Input
	SSLn0	Input	Input
	SSLn1 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISOn*4	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	CMOS output	Open-drain output
	MISOn	Input	Input

**Table 34.7 Relationship between pin states and bit settings (2 of 2)**

Mode	Pin	Pin state <sup>*2</sup>	
		PmnPFS.NCODR bit for I/O ports = 0	PmnPFS.NCODR bit for I/O ports = 1
Slave mode (clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKn	Input	Input
	SSLn0 to SSLn3 <sup>*5</sup>	Hi-Z <sup>*1</sup>	Hi-Z <sup>*1</sup>
	MOSIn	Input	Input
	MISO <sub>n</sub>	CMOS output	Open-drain output

Note 1. This function is not supported in this mode.

Note 2. SPI settings are not reflected in multiplexed pins for which the SPI function is not selected.

Note 3. Motorola-SPI: When SSLn0 is at the active level, the pin state is Hi-Z. Whether or not the input signal is at the active level determines the setting of the SPCR3.SSLOP bit.

TI-SSP: From when SSL0 is at the active level until communication is completed, the pin state is Hi-Z under the condition SPCR.SPE = 1.

Note 4. Motorola-SPI: When SSLn0 is at the non-active level or the SPCR.SPE bit is 0, the pin state is Hi-Z. Whether or not the input signal is at the active level determines the setting of the SPCR3.SSLOP bit.

TI-SSP: When SSL0 is except the communication period or when the SPE bit of SPCR is 0 (assertion after SPE = 1 and communication is completed), the pin status changes to Hi-Z.

Note 5. These pins are available for use as I/O port pins.

The SPI in single-master mode (SPI operation) or multi-master mode (SPI operation) determines the MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) based on the MOIFE and MOIFV bit settings in SPCR2, as listed in [Table 34.8](#).

**Table 34.8 MOSI signal value determination during SSL negation**

MOIFE bit	MOIFV bit	MOSIn signal value during SSL negation
0	0, 1	Final data from previous transfer
1	0	Low
1	1	High

### 34.3.3 SPI System Configuration Examples

This configuration example describes that 0 level of SSL<sub>n</sub> signals is active level.

When connecting and using in a multi-slave or multi-master mode, the transfer format of the connected device should be unified to either Motorola-SPI or TI-SSP.

#### 34.3.3.1 Single-master/single-slave with the MCU as a master

[Figure 34.6](#) shows a single-master/single-slave SPI system configuration example where the MCU is used as a master. In the single-master/single-slave configuration, the SSL<sub>n</sub> outputs of the MCU (master) are not used. The SSL input of the SPI slave is fixed to the low level, and the SPI slave is maintained in the selected state.<sup>\*1</sup>

Note 1. In the transfer format configured when the SPCMDm.CPHA bit is 0, the SSL signal for some slave devices cannot be fixed to an active level. In this case, always connect the SSL<sub>n</sub> output of the MCU to the SSL input of the slave device.

The MCU (master) drives the RSPCK<sub>n</sub> and MOSIn signals. The SPI slave drives the MISO signal.

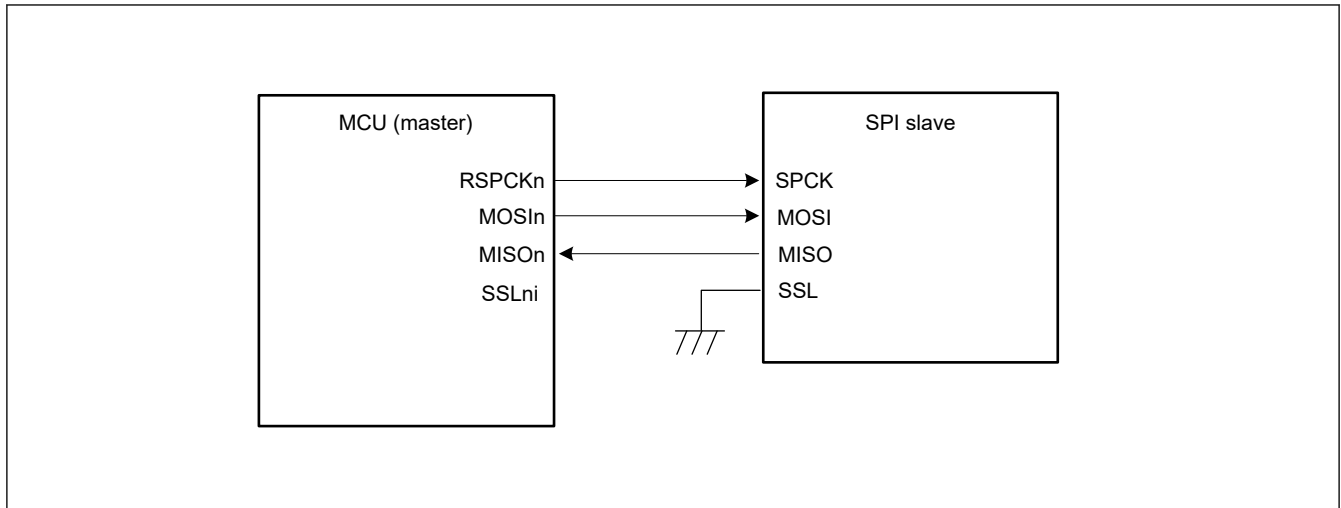


Figure 34.6 Single-master/single-slave configuration example with the MCU as a master

### 34.3.3.2 Single-master/single-slave with the MCU as a slave

Figure 34.7 shows a single-master/single-slave SPI system configuration example where the MCU is used as a slave. When the MCU operates as a slave, the SSLn0 pin is used as SSL input. The SPI master drives the RSPCK and MOSI signals. The MCU (slave) drives the MISO n signal.\*1

Note 1. When SSLn0 is at a non-active level, the pin state is Hi-Z.

In the single-slave configuration when the SPCMDm.CPHA bit is set to 1, the SPCR.SPFRF bit is set to 0, and SPCR.SPMS is set to 0, the SSLn0 input of the MCU (slave) is fixed to the low level and the MCU (slave) is maintained in the selected state. This enables serial transfer execution (Figure 34.8). However, the communication end interrupt does not output when SSL0 input is fixed as Figure 34.8.

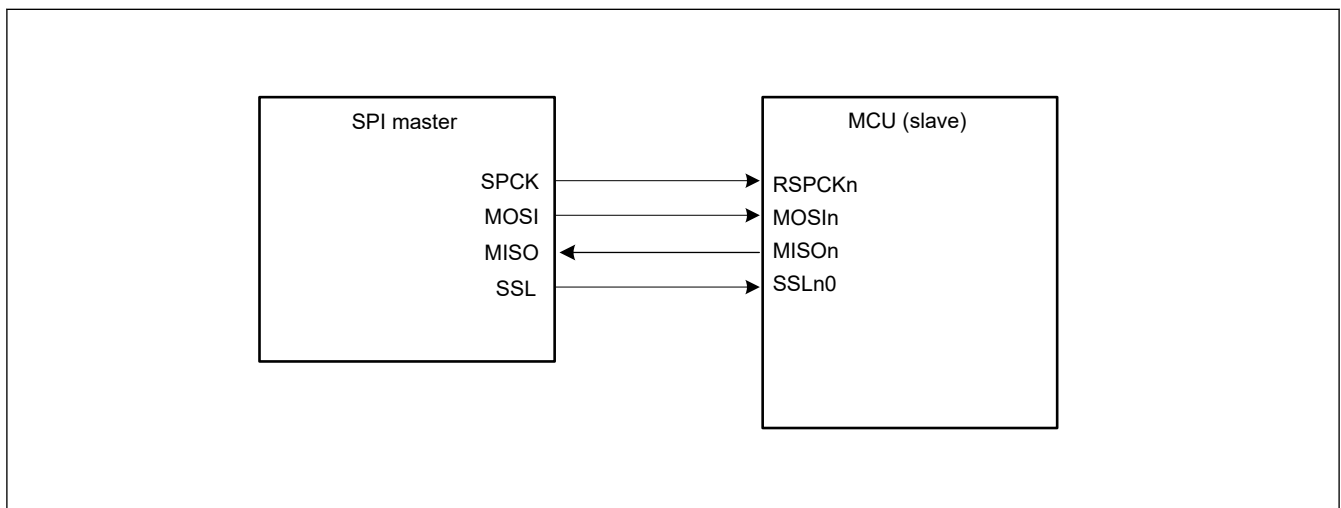
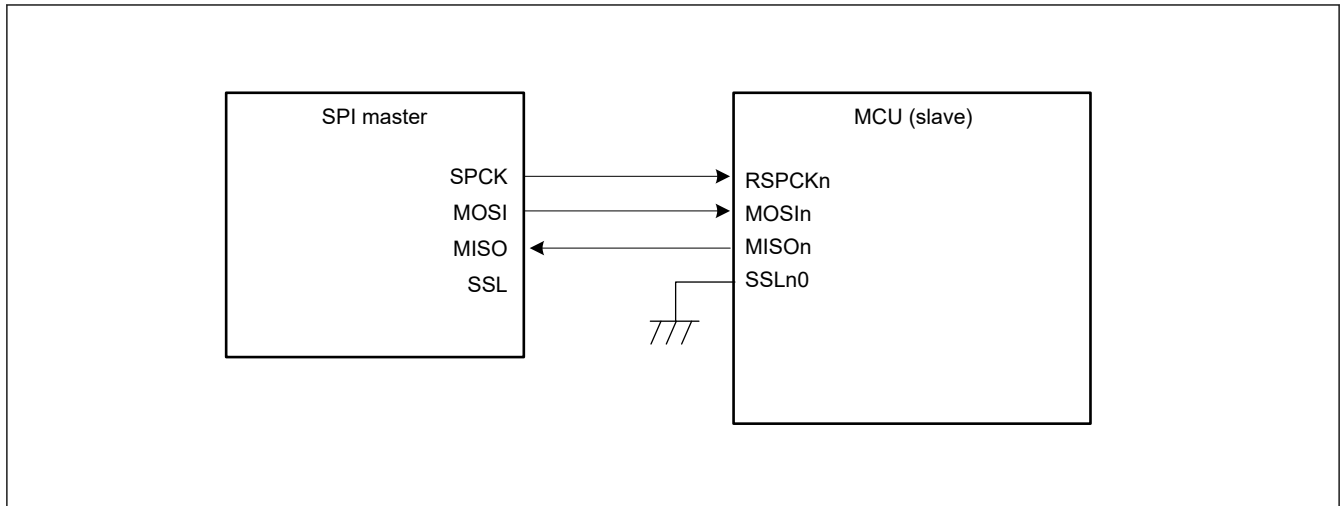


Figure 34.7 Single-master/single-slave configuration example with the MCU as a slave and CPHA = 0



**Figure 34.8** Single-master/single-slave configuration example with the MCU as a slave and CPHA = 1

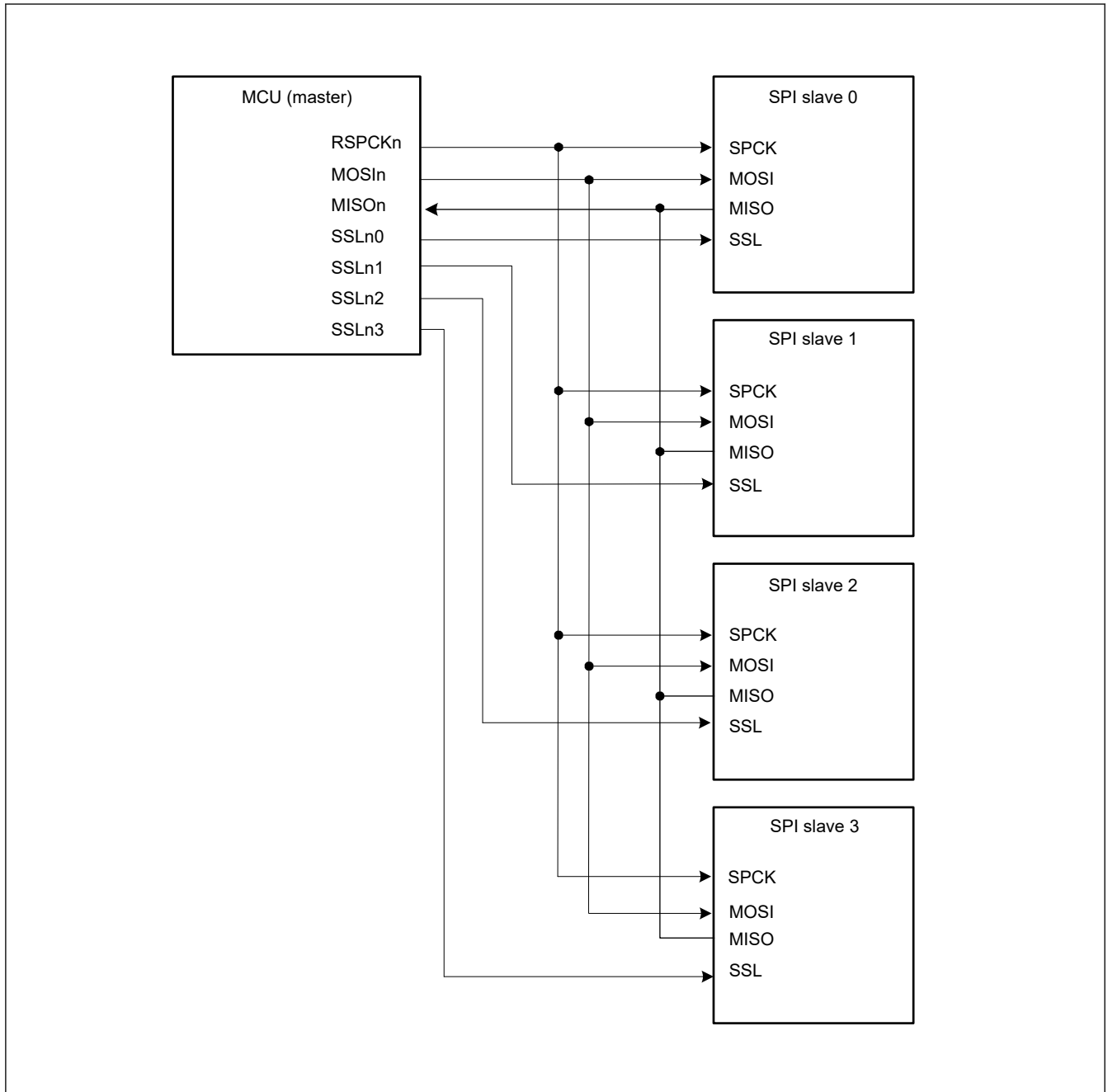
### 34.3.3.3 Single-master/multi-slave with the MCU as a master

Figure 34.9 shows a single-master/multi-slave SPI system configuration example where the MCU is used as a master. In this example, the SPI system includes the MCU (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCKn and MOSIn outputs of the MCU (master) are connected to the RSPCK and MOSI inputs of SPI slaves 0 to 3. The MISO outputs of SPI slaves 0 to 3 are all connected to the MISO<sub>n</sub> input of the MCU (master). The SSLn0 to SSLn3 outputs of the MCU (master) are connected to the SSL inputs of SPI slaves 0 to 3, respectively.

The MCU (master) drives the RSPCK<sub>n</sub>, MOSIn, and SSLn0 to SSLn3 signals. Out of the SPI slaves 0 to 3, the slave that receives low-level input into the SSL input drives the MISO signal.





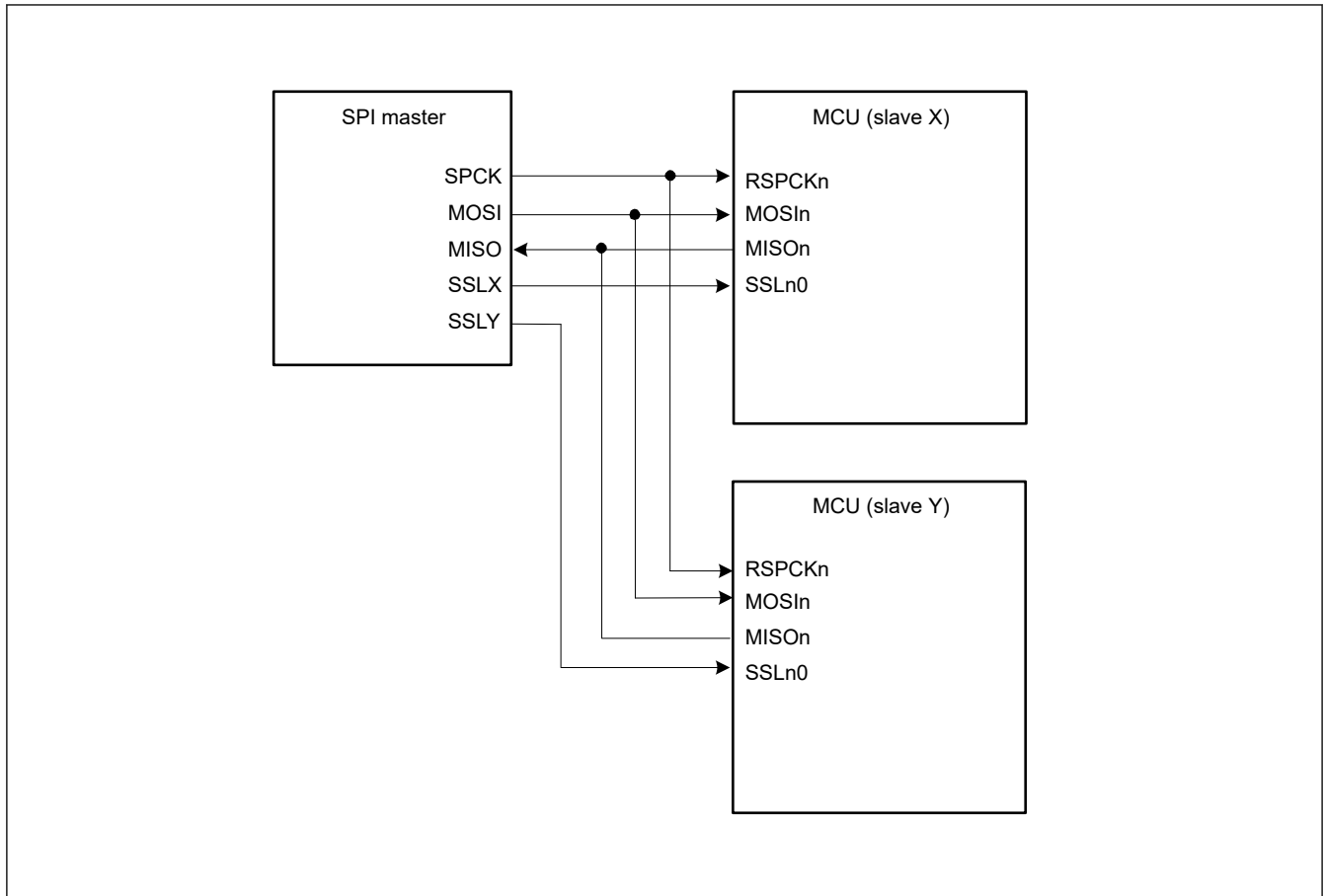
**Figure 34.9** Single-master/multi-slave configuration example with the MCU as a master

### 34.3.3.4 Single-master/multi-slave with the MCU as a slave

Figure 34.10 shows a single-master/multi-slave SPI system configuration example where the MCU is used as a slave. In this example, the SPI system includes an SPI master and two MCUs (slaves X and Y).

The SPCK and MOSI outputs of the SPI master are connected to the RSPCKn and MOSIn inputs of the MCUs (slaves X and Y). The MISO outputs of the MCUs (slaves X and Y) are all connected to the MISO input of the SPI master. The SSLX and SSLY outputs of the SPI master are connected to the SSLn0 inputs of the MCUs (slaves X and Y, respectively).

The SPI master drives the SPCK, MOSI, SSLX, and SSLY signals. Of the MCUs (slaves X and Y), the slave that receives low-level input into the SSLn0 input drives the MISO signal.



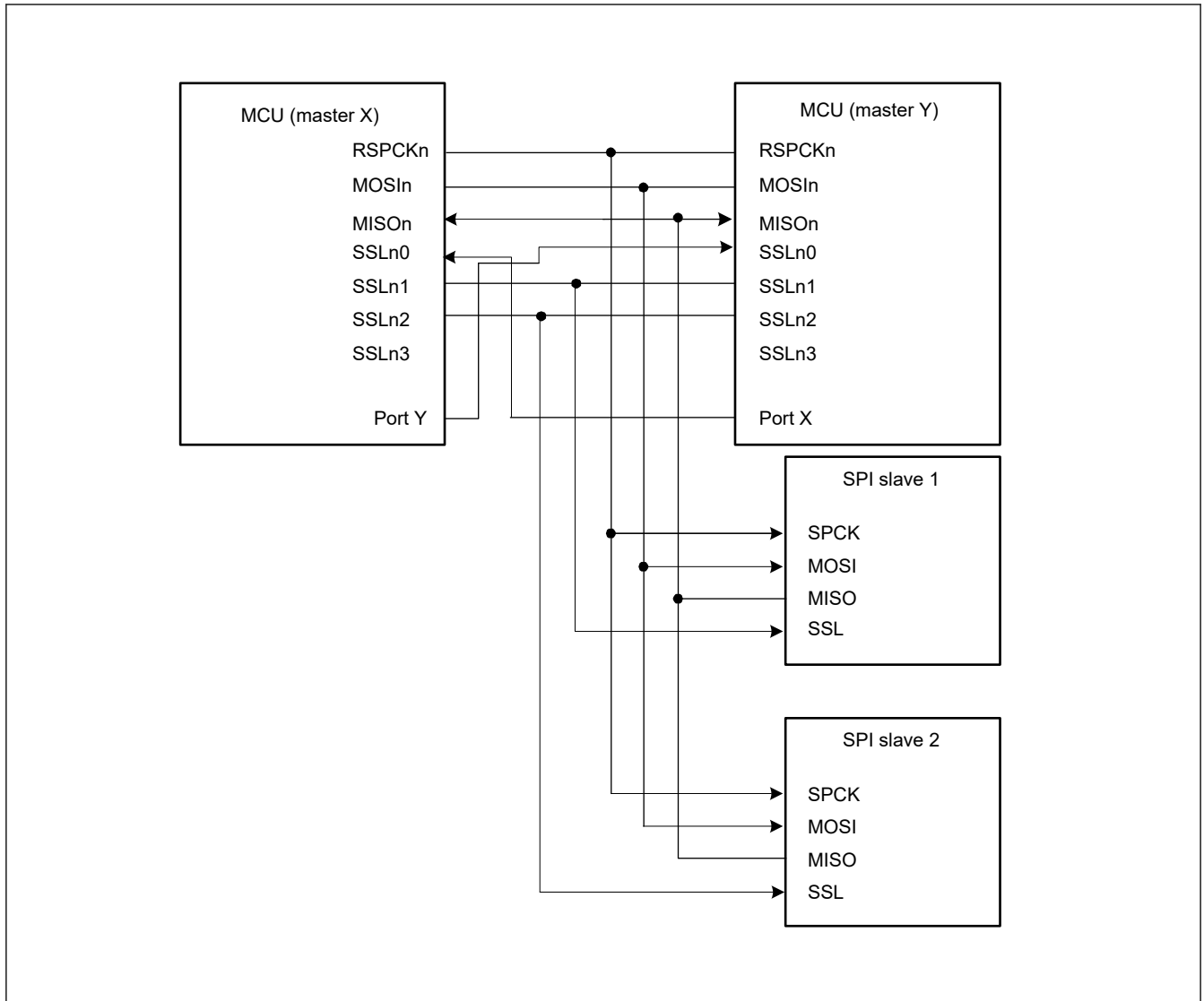
**Figure 34.10** Single-master/multi-slave configuration example with the MCU as a slave

### 34.3.3.5 Multi-master/multi-slave with the MCU as a master

[Figure 34.11](#) shows a multi-master/multi-slave SPI system configuration example where the MCU is used as a master. In this example, the SPI system includes two MCUs (masters X and Y) and two SPI slaves (SPI slaves 1 and 2).

The RSPCKn and MOSIn outputs of the MCUs (masters X and Y) are connected to the RSPCK and MOSI inputs of SPI slaves 1 and 2. The MISO outputs of SPI slaves 1 and 2 are connected to the MISOOn inputs of the MCUs (masters X and Y). Any generic port Y output from the MCU (master X) is connected to the SSLn0 input of the MCU (master Y). Any generic port X output of the MCU (master Y) is connected to the SSLn0 input of the MCU (master X). The SSLn1 and SSLn2 outputs of the MCUs (masters X and Y) are connected to the SSL inputs of the SPI slaves 1 and 2. In this configuration example, because the system can be comprised solely of SSLn0 input, and SSLn1 and SSLn2 outputs for slave connections, the SSLn3 output of the MCU is not required.

The MCU drives the RSPCKn, MOSIn, SSLn1, and SSLn2 signals when the SSLn0 input level is high. When the SSLn0 input level is low, the MCU detects a mode fault error, sets RSPCKn, MOSIn, SSLn1, and SSLn2 to Hi-Z, and releases the SPI bus directly to the other master. Of the SPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives the MISO signal.



**Figure 34.11 Multi-master/multi-slave configuration example with the MCU as a master**

When setting TI SSP, enter the following levels for port X and port Y.

- Start of communication: the value of SPCR3.SSL0P of the other master.
- End of communication: the inverted value of SPCR3.SSL0P of the other master.

### 34.3.3.6 Master and slave in clock synchronous mode with the MCU configured as a master

Figure 34.12 shows a master and slave in clock synchronous mode configuration example where the MCU is used as a master. In this configuration, SSLn<sub>i</sub> of the MCU (master) are not used.

The MCU (master) drives the RSPCK<sub>n</sub> and MOSIn signals. The SPI slave drives the MISO signal.

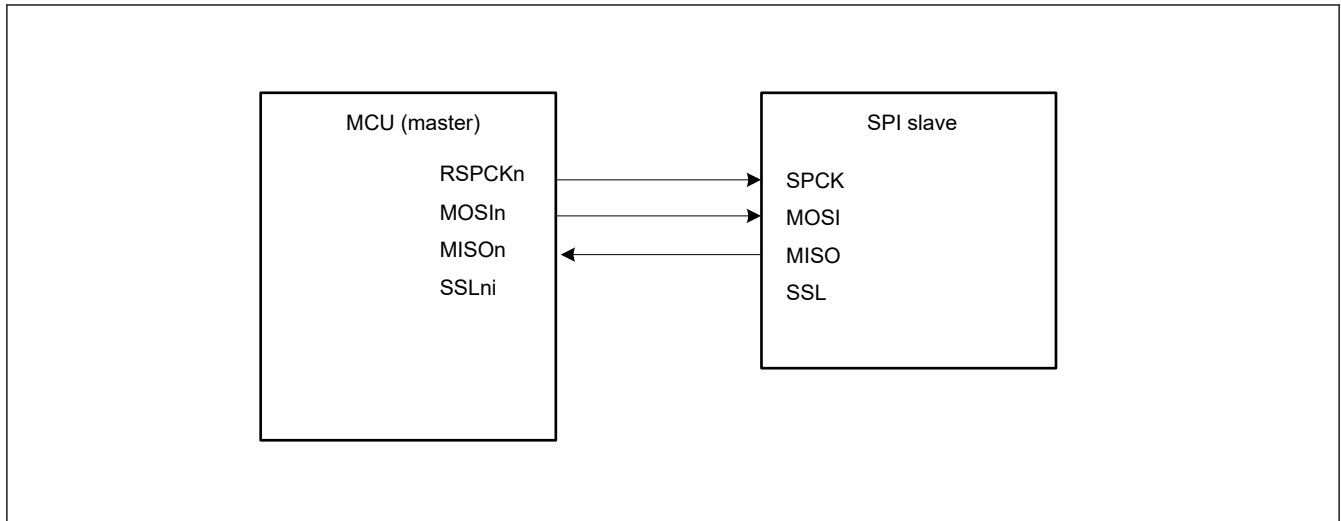


Figure 34.12 Clock synchronous master/slave configuration example with the MCU as a master

### 34.3.3.7 Master and slave in clock synchronous mode with the MCU as a slave

Figure 34.13 shows a master and slave in clock synchronous mode configuration example where the MCU is used as a slave. When the MCU operates as a slave (clock synchronous operation), the MCU (slave) drives the MISOOn signal and the SPI master drives the SPCK and MOSI signals. In addition, SSLn0 to SSLn3 of the MCU (slave) are not used.

The MCU (slave) can only execute serial transfers in the single-slave configuration when the SPCMDm.CPHA bit is set to 1.

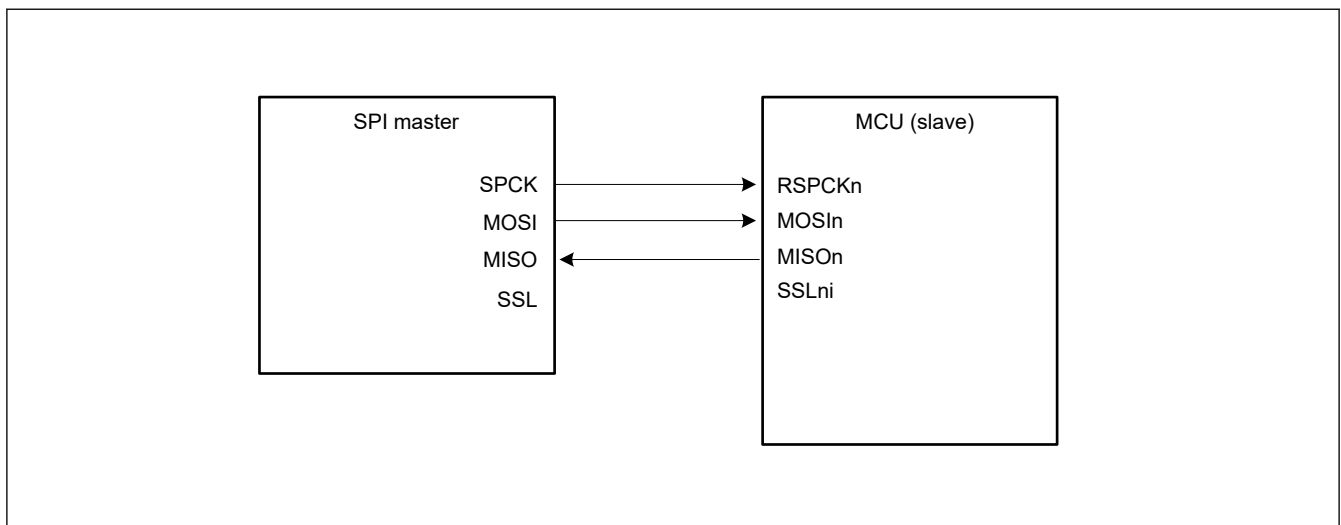


Figure 34.13 Clock synchronous master/slave configuration example with the MCU as a slave and CPHA = 1

### 34.3.4 Data Formats

The data format of the SPI depends on the settings in SPI Command Register  $m$  (SPCMDm) and the parity enable bit in SPI Control Register (SPCR.SPPE). Regardless of whether the MSB or LSB is first, the SPI treats the range from the LSB bit in the SPI Data Register (SPDR) to the bit associated with the selected data length, as transfer data.

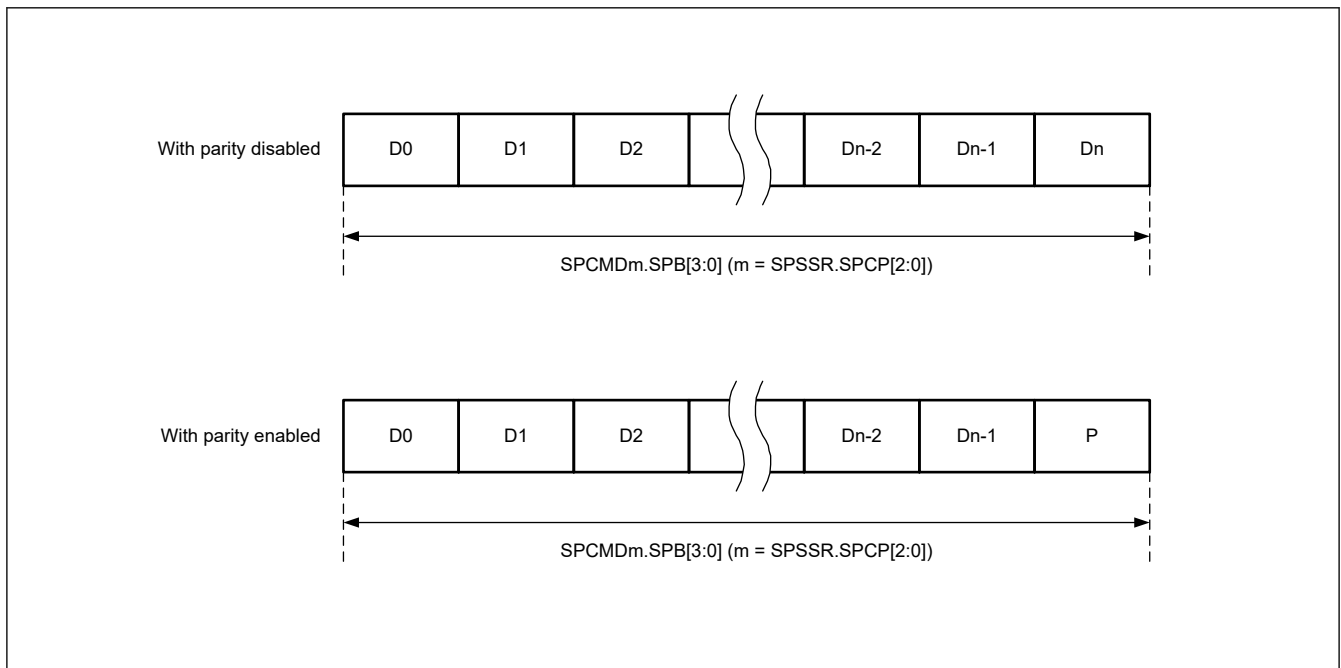
This section shows the format of one frame of data before or after transfer.

#### Data format with parity disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register  $m$  (SPCMDm.SPB[4:0]).

### Data format with parity enabled

When parity is enabled, transmission or reception of data proceeds with the length in bits selected in the SPI data length setting in SPI Command Register  $m$  (SPCMD $m$ .SPB[4:0]). In this case, however, the last bit is a parity bit.



**Figure 34.14** Data format with parity disabled and enabled

#### 34.3.4.1 Operation when parity is disabled (SPCR.SPPE = 0)

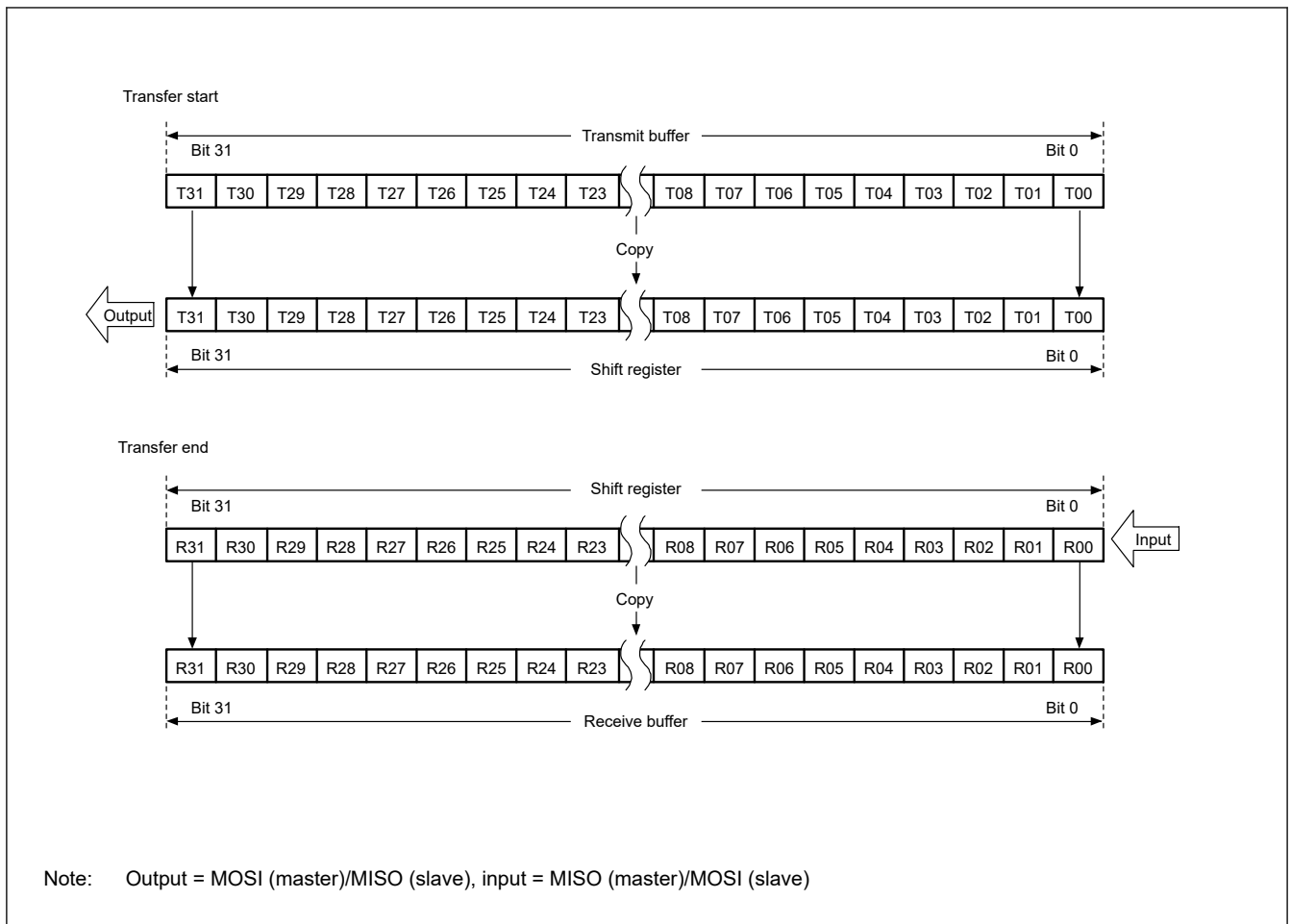
When parity is disabled, data for transmission is copied to the shift register with no pre-processing. This section describes the connection between the SPI Data Register (SPDR) and the shift register in terms of the combination of MSB- or LSB-first order and data length.

##### (1) MSB-first transfer with 32-bit data

Figure 34.15 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, a SPI data length of 32 bits, and MSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T31 to T30, and continuing to T00.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R31 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



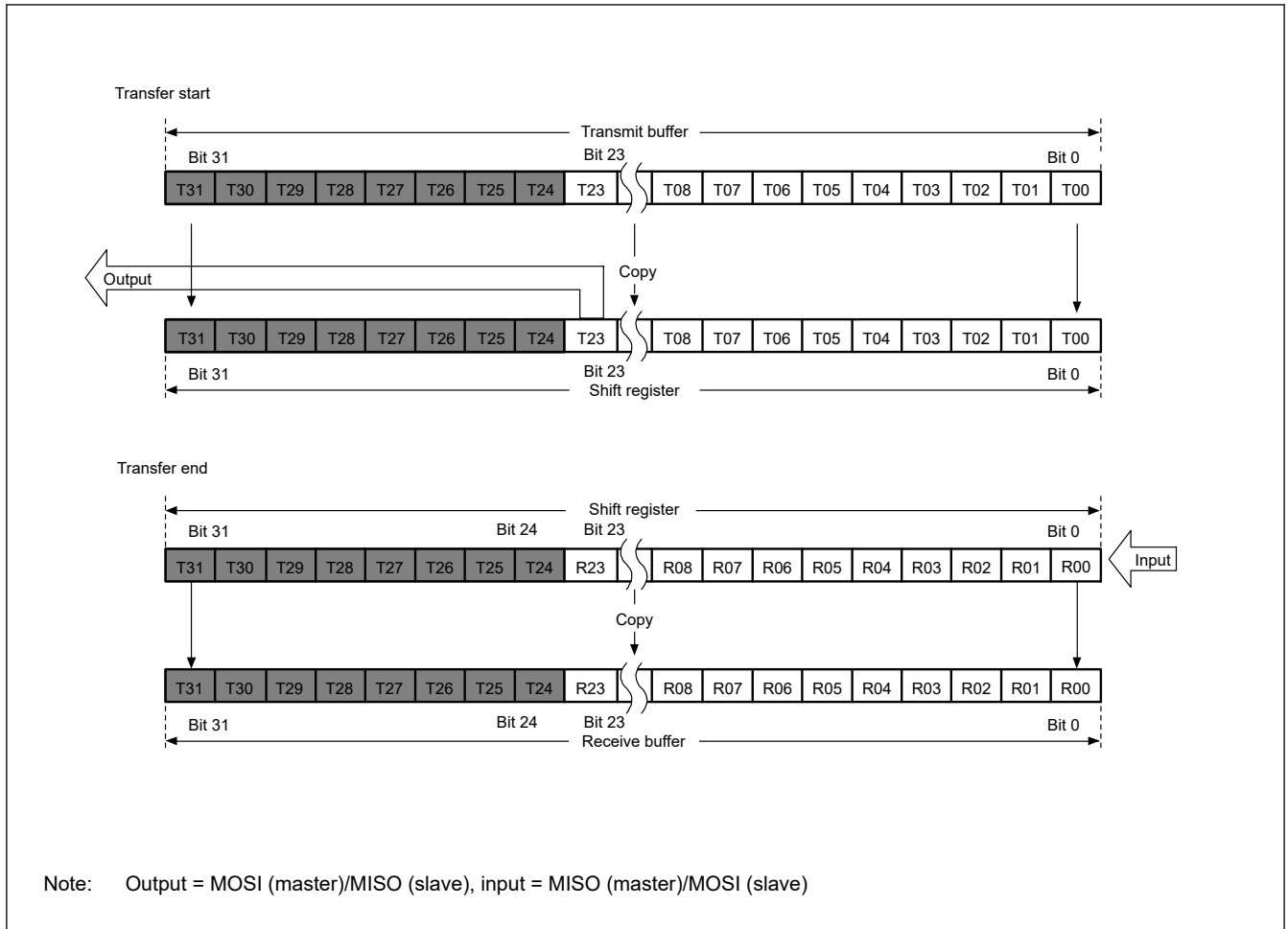
**Figure 34.15 MSB-first transfer with 32-bit data and parity disabled**

(2) MSB-first transfer with 24-bit data

Figure 34.16 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 24 bits for an example that is not 32 bits, and MSB-first selected.

In transmission, the lower 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission is shifted out from the shift register from T23 to T22, and continuing to T00.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R23 to R00 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



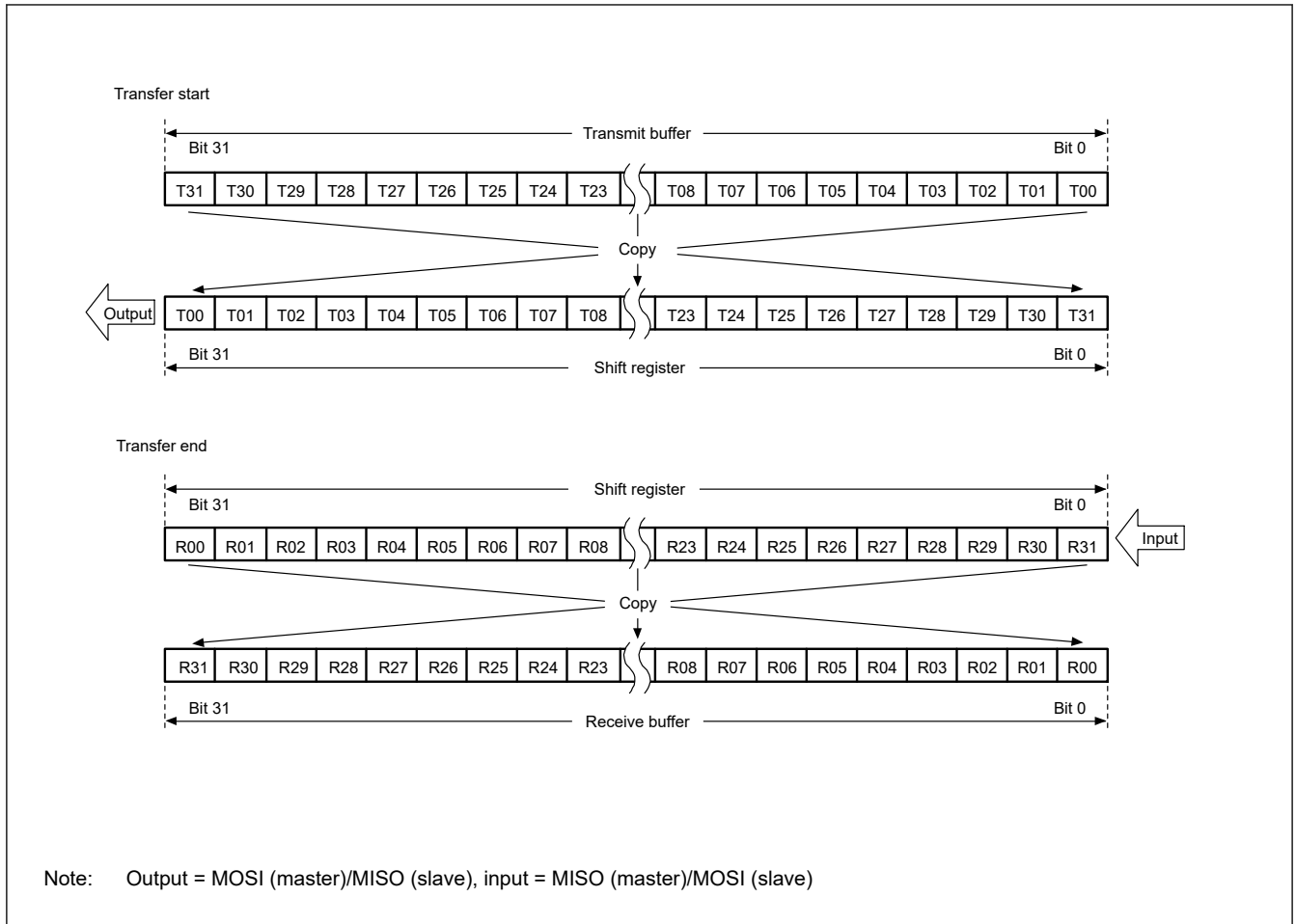
**Figure 34.16 MSB-first transfer with 24-bit data and parity disabled**

(3) LSB-first transfer with 32-bit data

Figure 34.17 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity disabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission is shifted out from the shift register in order from T00 to T01, and continuing to T31.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R00 to R31 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



**Figure 34.17 LSB-first transfer with 32-bit data and parity disabled**

(4) LSB-first transfer with 24-bit data

Figure 34.18 shows the operation of the SPI Data Register (SPDR) and the shift register in transfers with parity disabled, an SPI data length of 24 bits for an example that is not 32, and LSB-first selected.

In transmission, the lower 24 bits (T23 to T0) from the current stage of the transmit buffer are reordered bit-by-bit to obtain the order T0 to T23 for copying to the shift register. Data for transmission is shifted out from the shift register from T00 to T01, and continuing to T23.

In reception, received data is shifted in bit-by-bit through bit[8] of the shift register. When the R00 to R23 bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer.



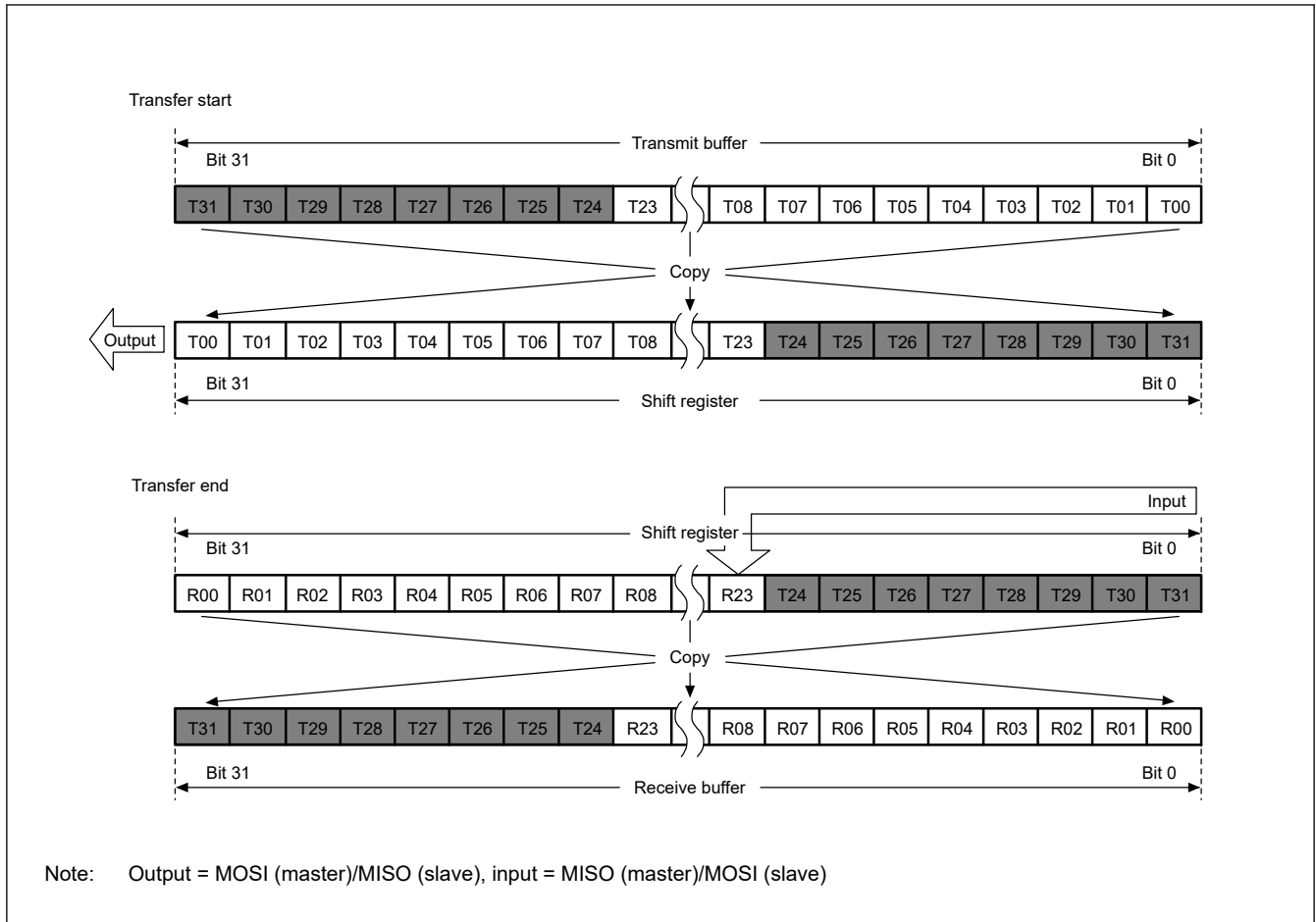


Figure 34.18 LSB-first transfer with 24-bit data and parity disabled

### 34.3.4.2 Operation when parity is enabled (SPCR.SPPE = 1)

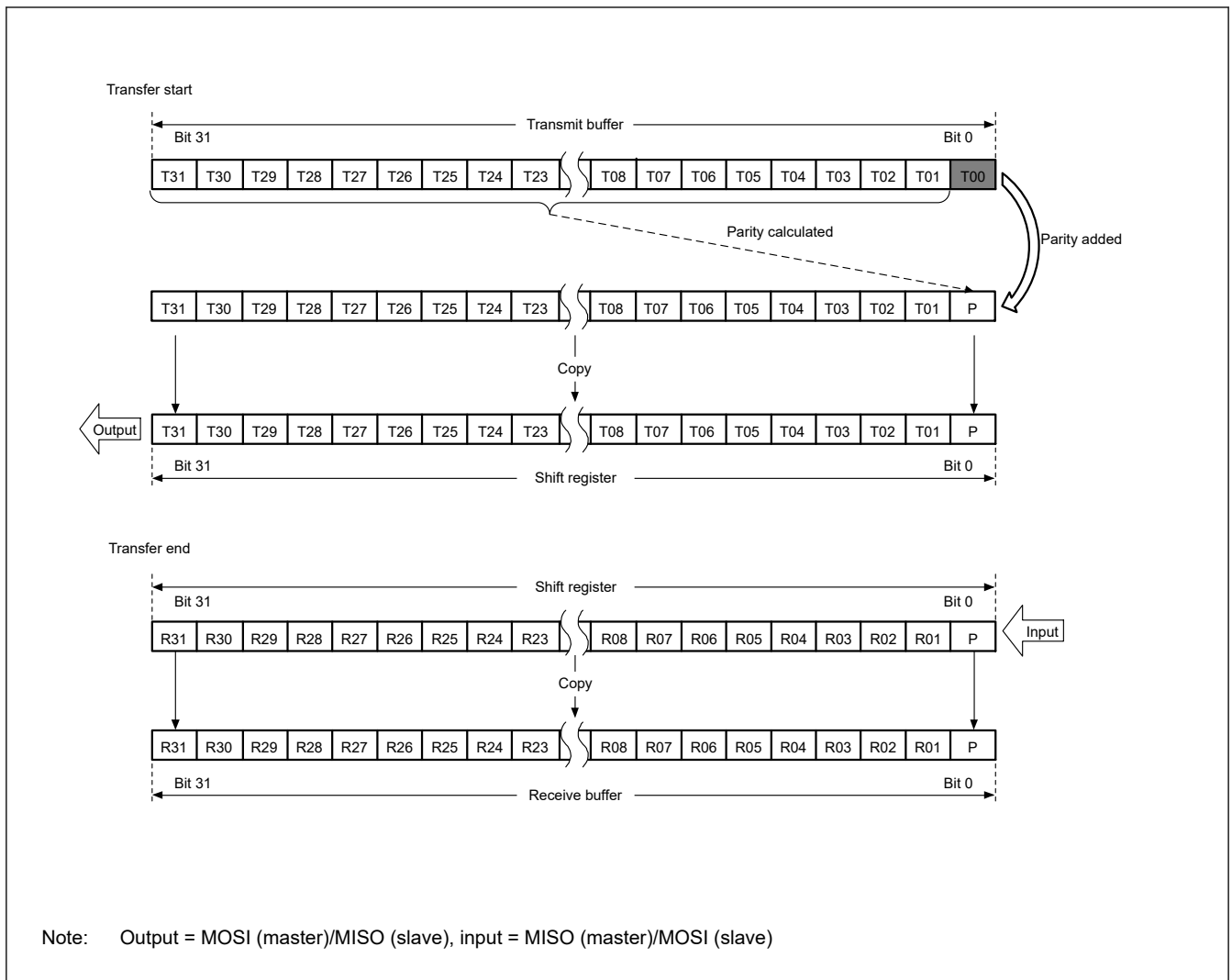
When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

#### (1) MSB-first transfer with 32-bit data

Figure 34.19 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 32 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole value is copied to the shift register. Data is transmitted in the order T31, T30, ..., T01, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R31 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P is checked for parity.



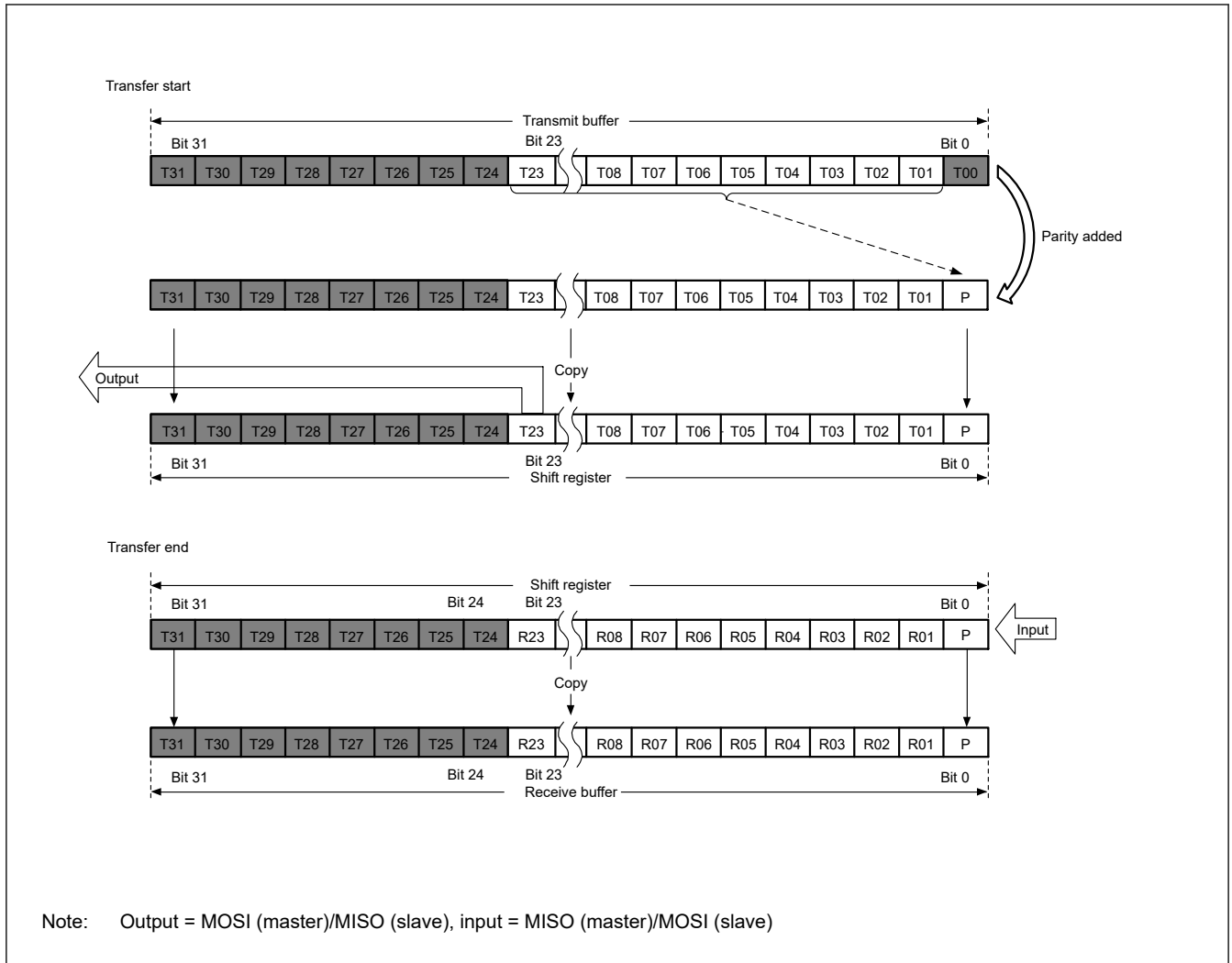
**Figure 34.19 MSB-first transfer with 32-bit data and parity enabled**

(2) MSB-first transfer with 24-bit data

Figure 34.20 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, a SPI data length of 24 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole value is copied to the shift register. Data is transmitted in the order T23, T22, ..., T01, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R23 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P is checked for parity.



**Figure 34.20 MSB-first transfer with 24-bit data and parity enabled**

(3) LSB-first transfer with 32-bit data

Figure 34.21 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, an SPI data length of 32 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole value is copied to the shift register. Data is transmitted in the order T00, T01, ..., T30, and P.

In reception, received data is shifted in bit-by-bit through bit[0] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P is checked for parity.

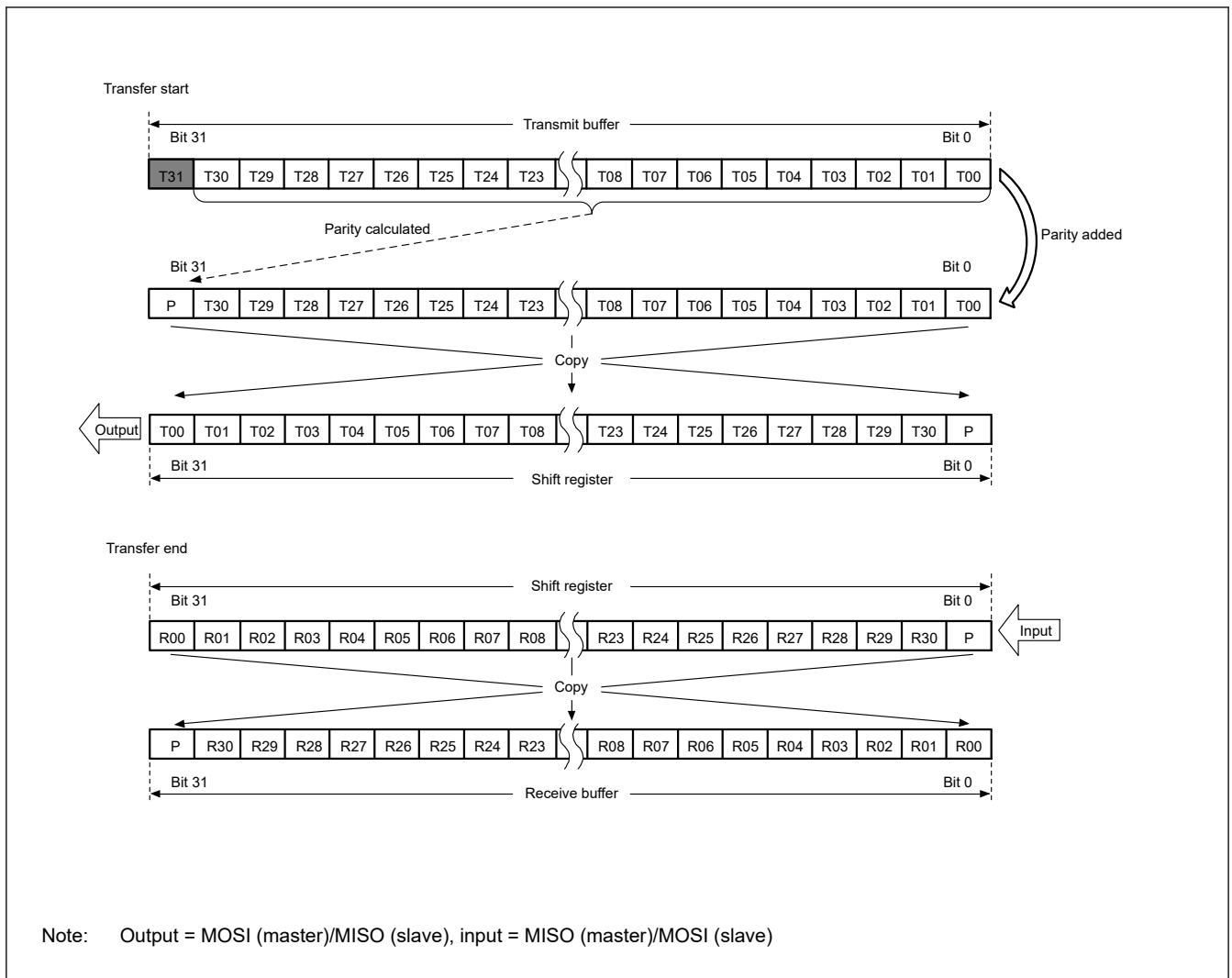


Figure 34.21 LSB-first transfer with 32-bit data and parity enabled

(4) LSB-first transfer with 24-bit data

Figure 34.22 shows the operation of the SPI Data Register (SPDR) and the shift register in a transfer with parity enabled, a SPI data length of 24 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T22 to T0. This replaces the final bit, T23, and the whole value is copied to the shift register. Data is transmitted in the order T00, T01, ..., T22, and P.

In reception, received data is shifted in bit-by-bit through bit[8] of the shift register. When the R00 to P bits are collected after input of the required number of RSPCK cycles, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P is checked for parity.

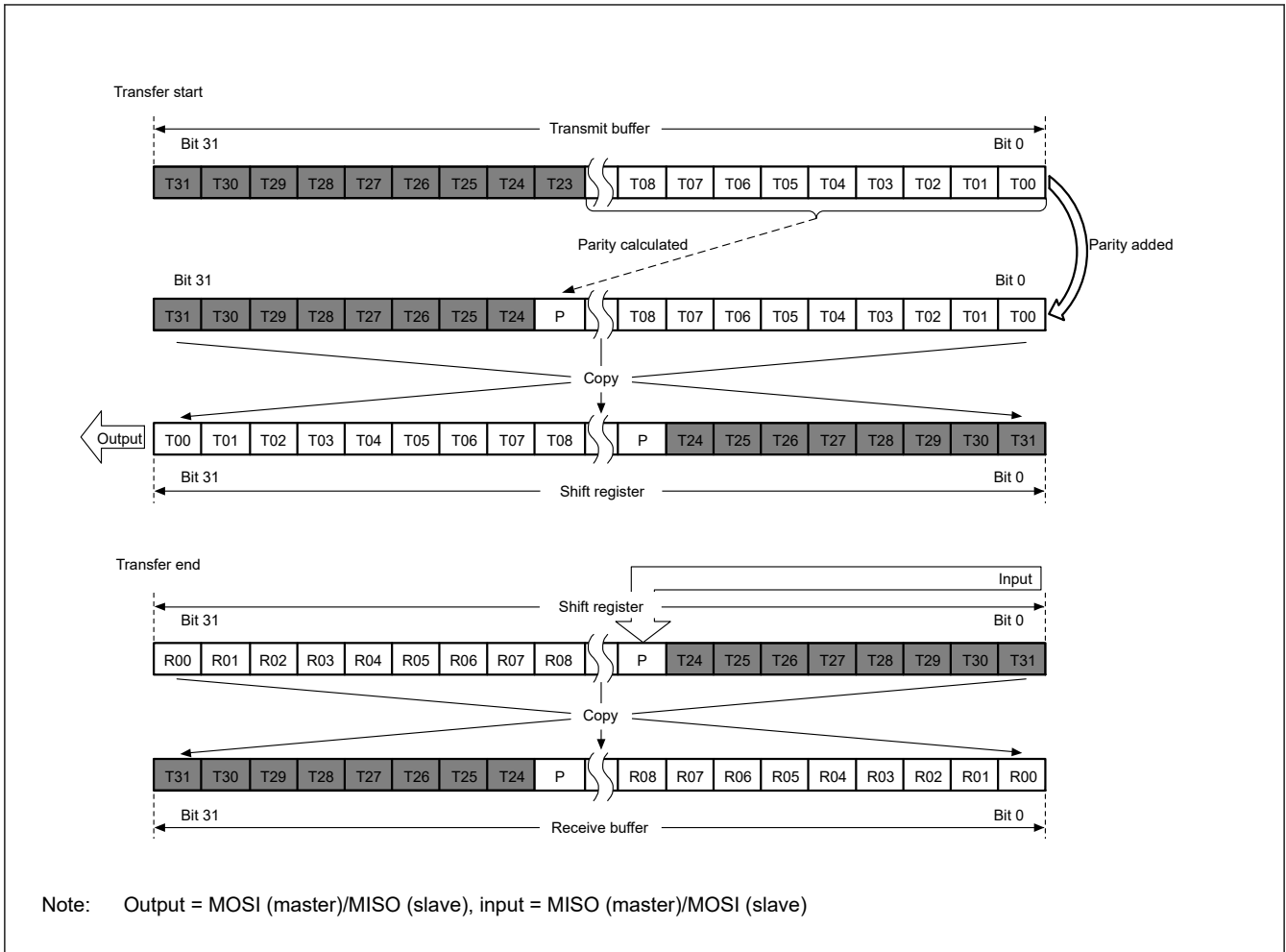


Figure 34.22 LSB-first transfer with 24-bit data and parity enabled

### 34.3.4.3 Byte Swap Transmission

When byte swapping is enabled, the data in the transmission buffer, swapped in 8-bit units, is copied to the shift register. Figure 34.23 shows the relationship between the SPDR (transmit buffer) and the shift register when transferring data with a 32-bit data length, using a combination of MSB / LSB first and with / without byte swap.

#### (1) MSB-first transfer. (When the byte swap is disabled.)

Data (Byte3 [T31 to T24] to Byte0 [T07 to T00]) in the transmit buffer are copied to the shift register. Bit values in the shift register are shifted and transmitted in the order of T31 → T30 → ... → T00 as transmit data.

#### (2) MSB-first transfer. (When the byte swap is enabled.)

Byte values of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte0 [T07 to T00] to Byte3 [T31 to T24]. Bit values in the shift register are shifted and transmitted in the order of T07 → T06 → ... → T00 → T15 → T14 → ... → T08 → T23 → T22 → ... → T16 → T31 → T30 → ... → T24 as transmit data.

#### (3) LSB-first transfer. (When the byte swap is disabled.)

Bit values of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte0 [T00 to T07] to Byte3 [T24 to T31]. Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → ... → T31 as transmit data.

(4) LSB-first transfer. (When the byte swap is enabled.)

Bit values of each byte of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte3 [T24 to T31] to Byte0 [T00 to T07].

Bit values in the shift register are shifted and transmitted in the order of T24 → T25 → ... → T31 → T16 → T17 → ... → T23 → T08 → T09 → ... → T15 → T00 → T01 → ... → T07 as transmit data.

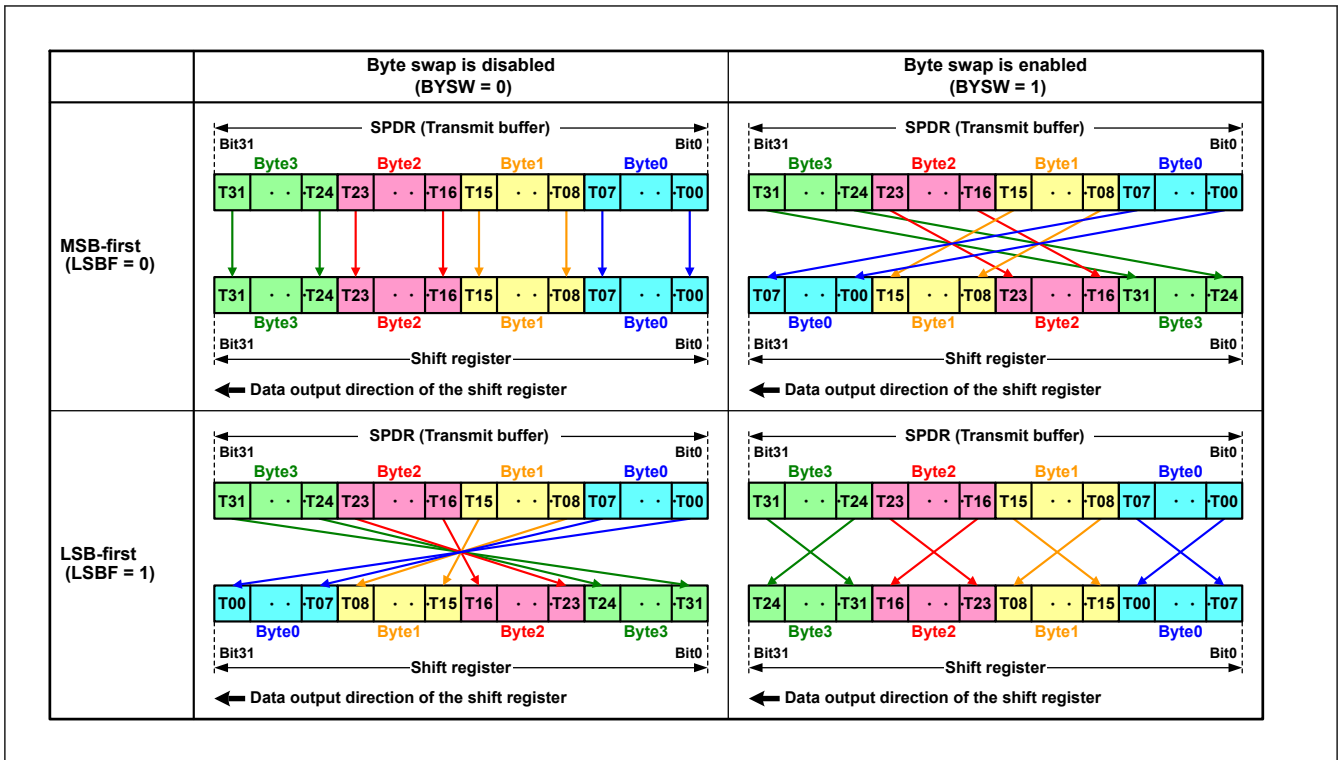


Figure 34.23 Byte swap with MSB/LSB transfer (32bit)

Figure 34.24 shows the relationship between the SPDR (transmit buffer) and the shift register when transferring data with a 16-bit data length, using a combination of MSB / LSB first and with / without byte swap.

1. MSB-first transfer. (When the byte swap is disabled.)  
Data (Byte1 [T15 to T08] to Byte0 [T07 to T00]) in the transmit buffer are copied to the shift register in the order of Byte1 [T15 to T08] to Byte0 [T07 to T00], Byte1 [T15 to T08] to Byte0 [T07 to T00]. Bit values in the shift register are shifted and transmitted in the order of T15 → T14 → ... T00 as transmit data.
2. MSB-first transfer. (When the byte swap is enabled.)  
Byte values of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte0 [T07 to T00] to Byte1 [T15 to T08], Byte0 [T07 to T00] to Byte1 [T15 to T08]. Bit values in the shift register are shifted and transmitted in the order of T07 → T06 → ... T00 → T15 → T14 → ... T08 as transmit data.
3. LSB-first transfer. (When the byte swap is disabled.)  
Bit values of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte0 [T00 to T07] to Byte1 [T08 to T15], Byte0 [T00 to T07] to Byte1 [T08 to T15]. Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → ... T15 as transmit data.
4. LSB-first transfer. (When the byte swap is enabled.)  
Bit values of each byte of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte1 [T08 to T15] to Byte0 [T00 to T07], Byte1 [T08 to T15] to Byte0 [T00 to T07]. Bit values in the shift register are shifted and transmitted in the order of T08 → T09 → ... T15 → T00 → T01 → ... T07 as transmit data.

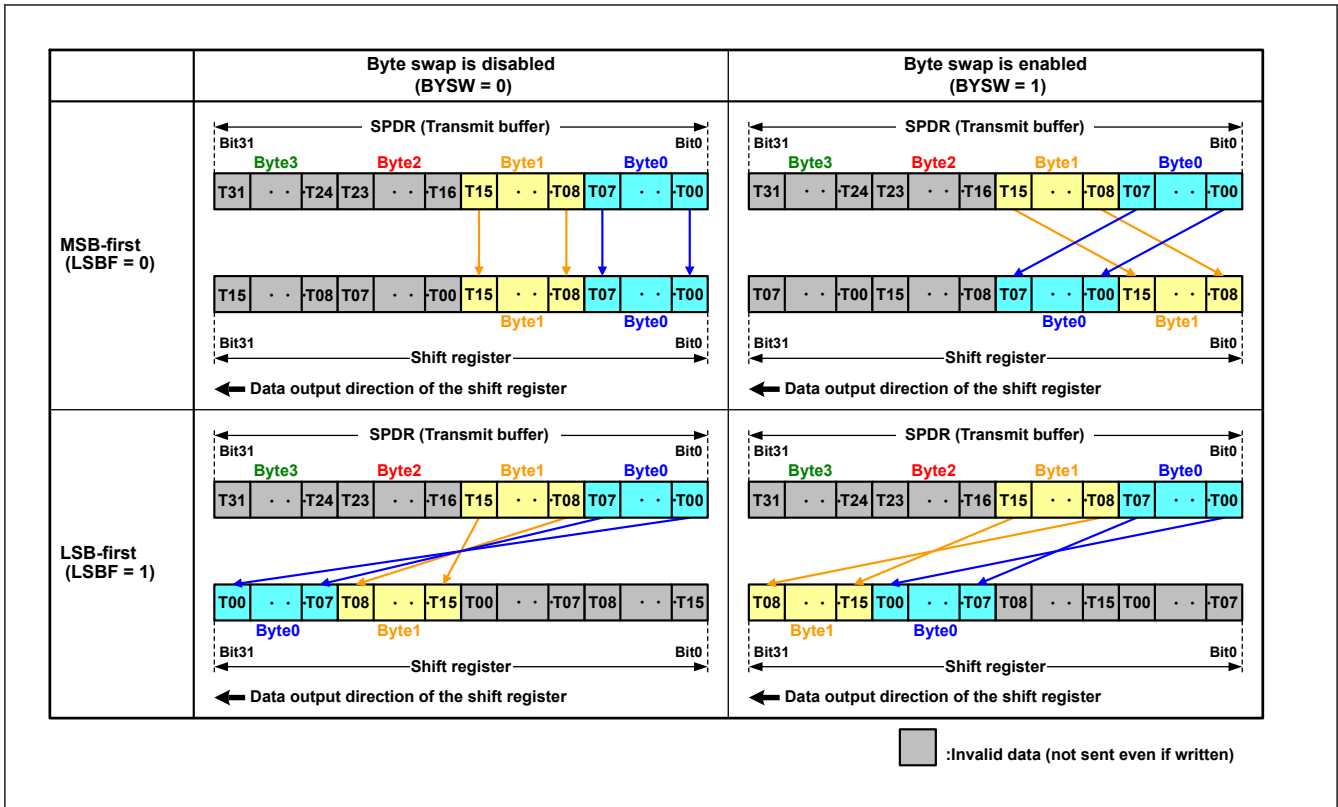


Figure 34.24 Byte swap with MSB/LSB transfer (16bit)

- Note:
1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SP[4:0] setting). If setting the other length, the behavior is not guaranteed.
  2. When the byte swap is valid, set the parity function as invalid (SPCR.SPPE bit = 0). If setting the parity function as valid (SPPE bit = 1), the behavior is not guaranteed.
  3. Set SPDCR.BYSW bit, when SPCR.SPE bit is 0. If rewriting BYSW bit, when SPE bit is 1, the behavior after it is not guaranteed.

### 34.3.4.4 Byte Swap Reception

When byte swap is enabled, the data in the shift register, swapped in 8-bit units, is copied to the receive buffer. Figure 34.25 shows the relationship between the shift register and SPDR (reception buffer) when transferring data with a 32-bit data length, using a combination of MSB / LSB first and with / without byte swap.

#### (1) MSB-first transfer (When the byte swap is disabled)

The first received data (R31) is stored in bit 0 of the shift register, and received data is shifted in the order of R31 → R30 → ... → R00.

When necessary RSPCK cycles are input and data is stored from Byte3 [R31 to R24] to Byte0 [R07 to R00], the shift register value is copied to the receive buffer.

#### (2) MSB-first transfer (When the byte swap is enabled)

The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07 → R06 → ... → R00 → R15 → R14 → ... → R08 → R23 → R22 → ... → R16 → R31 → R30 → ... → R24.

When necessary RSPCK cycles are input and data is stored from Byte0 [R07 to R00] to Byte3 [R31 to R24], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

#### (3) LSB-first transfer (When the byte swap is disabled)

The first received data (R00) is stored in bit 0 of the shift register, and received data is shifted in the order of R00 → R01 → ... → R31.

When necessary RSPCK cycles are input and data is stored from Byte0 [R00 to R07] to Byte3 [R24 to R31], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

(4) LSB-first transfer (When the byte swap is enabled)

The first received data (R24) is stored in bit 0 of the shift register, and received data is shifted in the order of R24 → R25 → ... → R31 → R16 → R17 → ... → R23 → R08 → R09 → ... → R15 → R00 → R01 → ... → R07.

When necessary RSPCK cycles are input and data is stored from Byte3 [R24 to R31] to Byte0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

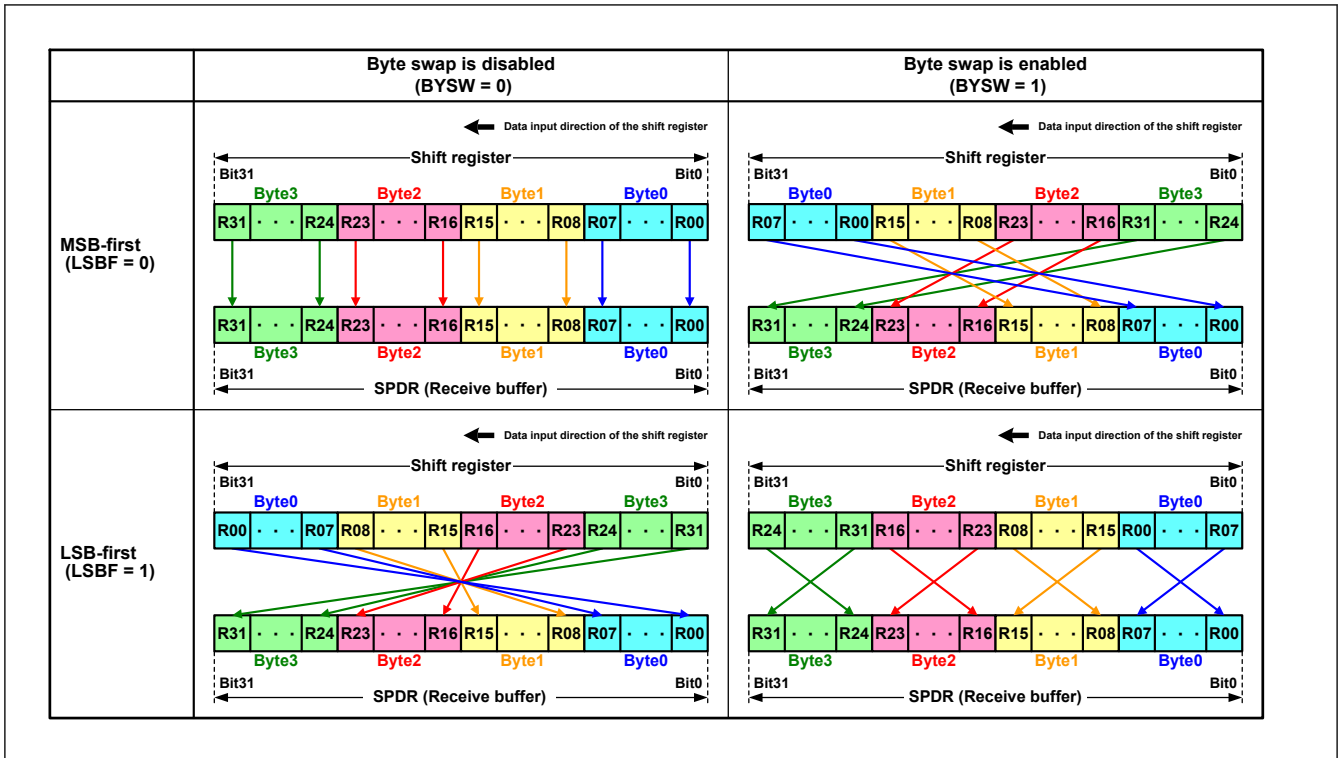


Figure 34.25 Byte swap with MSB/LSB transfer (32-bit)

Figure 34.26 shows the relationship between the shift register and SPDR (reception buffer) when transferring data with a 16-bit data length, using a combination of MSB / LSB first and with / without byte swap.

1. MSB-first transfer (when the byte swap is disabled)
 

The first received data (R15) is stored in bit 0 of the shift register, and received data is shifted in the order of R15 → R14 → ... R00. When necessary RSPCK cycles are input and data is stored from Byte3 [R31 to R24] to Byte0 [R07 to R00], the shift register value is copied to the receive buffer.
2. MSB-first transfer (when the byte swap is enabled)
 

The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07 → R06 → ... R00 → R15 → R14 → ... R08. When necessary RSPCK cycles are input and data is stored from Byte0 [R07 to R00] to Byte1 [R15 to R08], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].
3. LSB-first transfer (when the byte swap is disabled)
 

The first received data (R00) is stored in bit 15 of the shift register, and received data is shifted in the order of R00 → R01 → ... R07 → R08 → R09 → ... R15. When necessary RSPCK cycles are input and data is stored from Byte0 [R00 to R07] to Byte1 [R08 to R15], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07-R00].
4. LSB-first transfer (when the byte swap is enabled)
 

The first received data (R08) is stored in bit 15 of the shift register, and received data is shifted in the order of R08 → R09 → ... R15 → R00 → R01 → ... R07. When necessary RSPCK cycles are input and data is stored from Byte1 [R08



to R15] to Byte0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

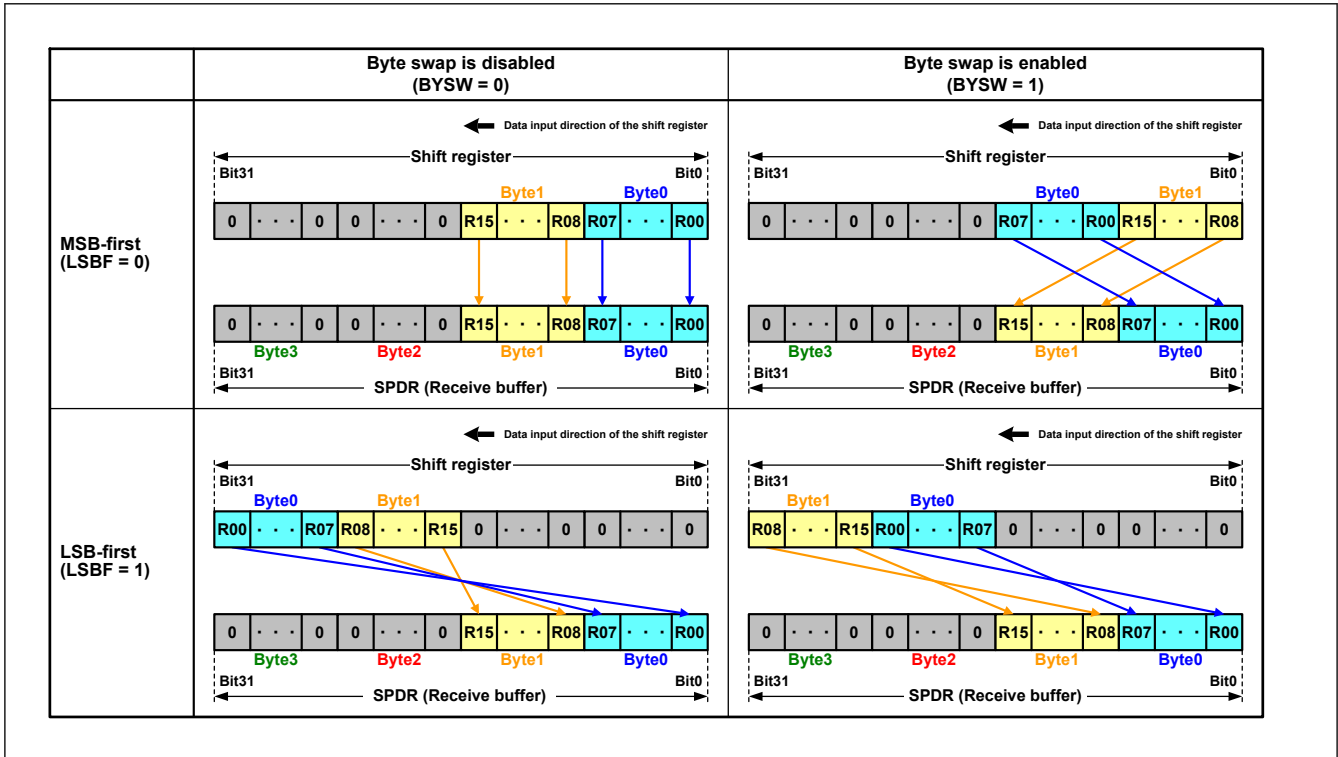


Figure 34.26 Byte swap with MSB/LSB transfer(16-bit)

- Note:
1. When using the byte swap, set 16 bits or 32 bits to the data length (SPCMDm.SPB[4:0] setting). If setting the other length, the behavior is not guaranteed.
  2. When the byte swap is valid, set the parity function as invalid (SPCR.SPPE bit = 0). If setting the parity function as valid (SPPE bit = 1), the behavior is not guaranteed.
  3. Set SPDCR.BYSW bit, when SPCR.SPE bit is 0. If rewriting BYSW bit, when SPE bit is 1, the behavior after it is not guaranteed.

### 34.3.5 Transfer Formats

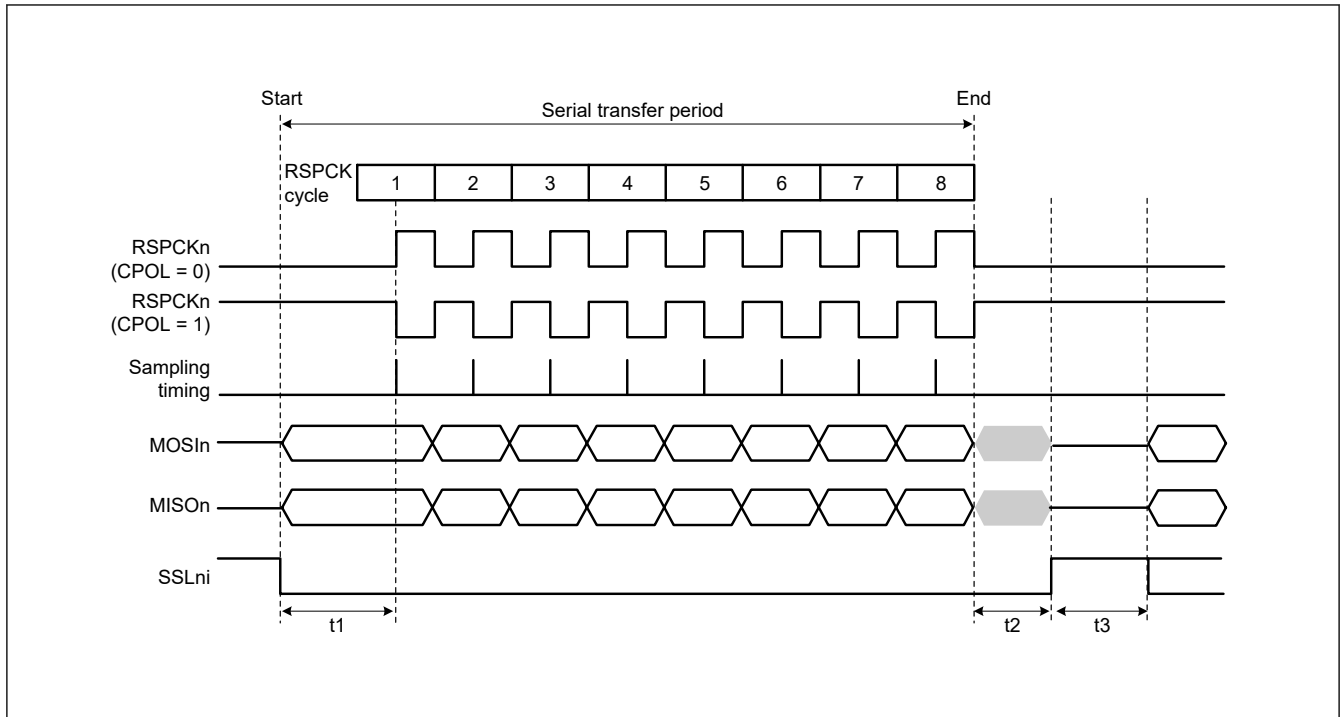
#### 34.3.5.1 When CPHA = 0

Figure 34.27 shows an example transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Do not perform clock synchronous operation (SPCR.SPMS = 1) when the SPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 34.27, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0, and RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI settings. For details, see section 34.3.2. Controlling the SPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIn and MISOOn signals begins at an SSLni signal assertion. The first RSPCKn signal change that occurs after the SSLni signal assertion becomes the first transfer data fetch. After this, data is sampled every 1 RSPCKn cycle. The change timing for MOSIn and MISOOn signals is 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCKn signal operation timing as it only affects the signal polarity.

t1 denotes the RSPCK delay, the period from an SSLni signal assertion to RSPCKn oscillation. t2 denotes the SSL negation delay, the period from the termination of RSPCKn oscillation to an SSLni signal negation. t3 denotes the next-access delay, the period in which SSLni signal assertion is suppressed for the next transfer after the end of serial transfer. t1, t2, and t3 are controlled by a master device running on the SPI system. For a description of t1, t2, and t3 when the SPI is in master mode, see section 34.3.12.1. Master mode operation.

[In the Motorola-SPI case]



**Figure 34.27 SPI transfer format when CPHA = 0, SPFRF = 0**

[In TI-SSP case]

Not supported in CPHA = 0

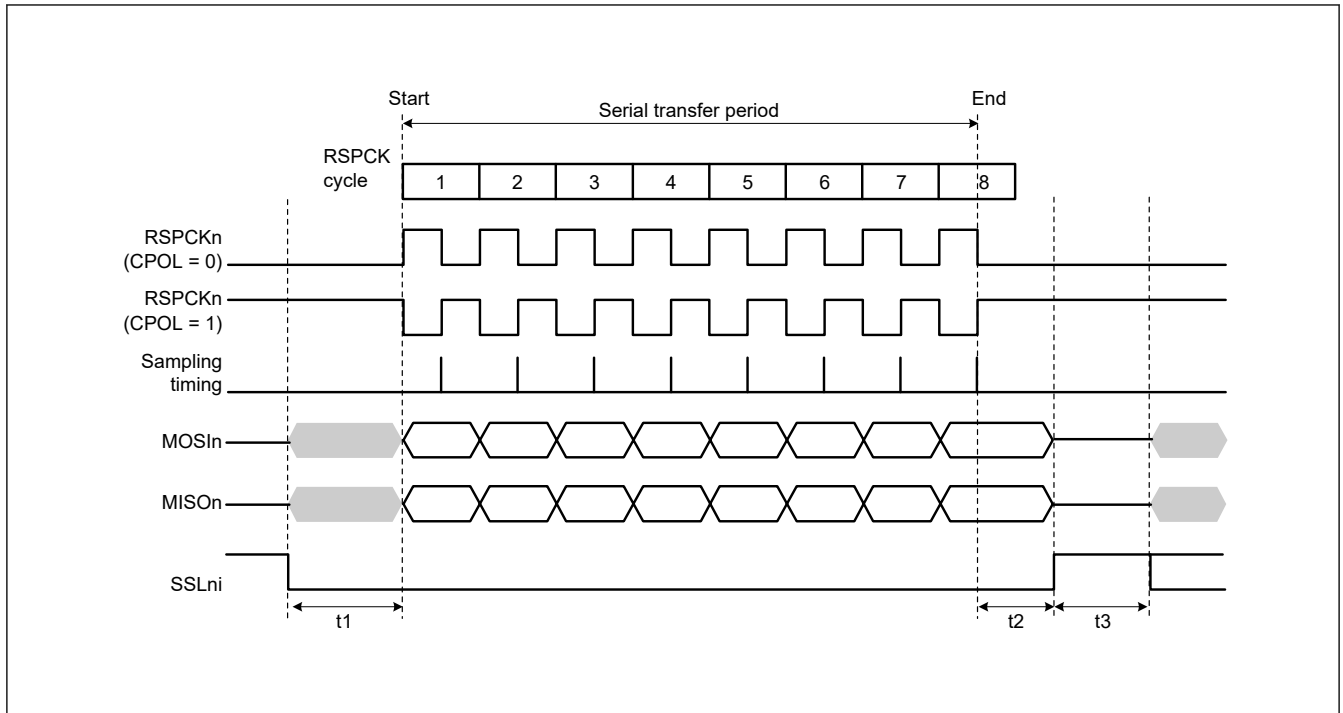
### 34.3.5.2 When CPHA = 1

Figure 34.28 shows an example transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLni signals are not used, and only the three signals RSPCKn, MOSIn, and MISO handle communications. In Figure 34.28, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0 and RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the SPI fetches serial transfer data into the shift register. The I/O directions of the signals depend on the SPI mode (master or slave mode). For details, see section 34.3.2. Controlling the SPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISO signal begins at an SSLni signal assertion. The output of valid data to the MOSIn and MISO signals begins at the first RSPCKn signal change that occurs after the SSLni signal assertion. After this, data is updated every 1 RSPCK cycle. The transfer data fetch timing is 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKn signal operation timing. It only affects the signal polarity.

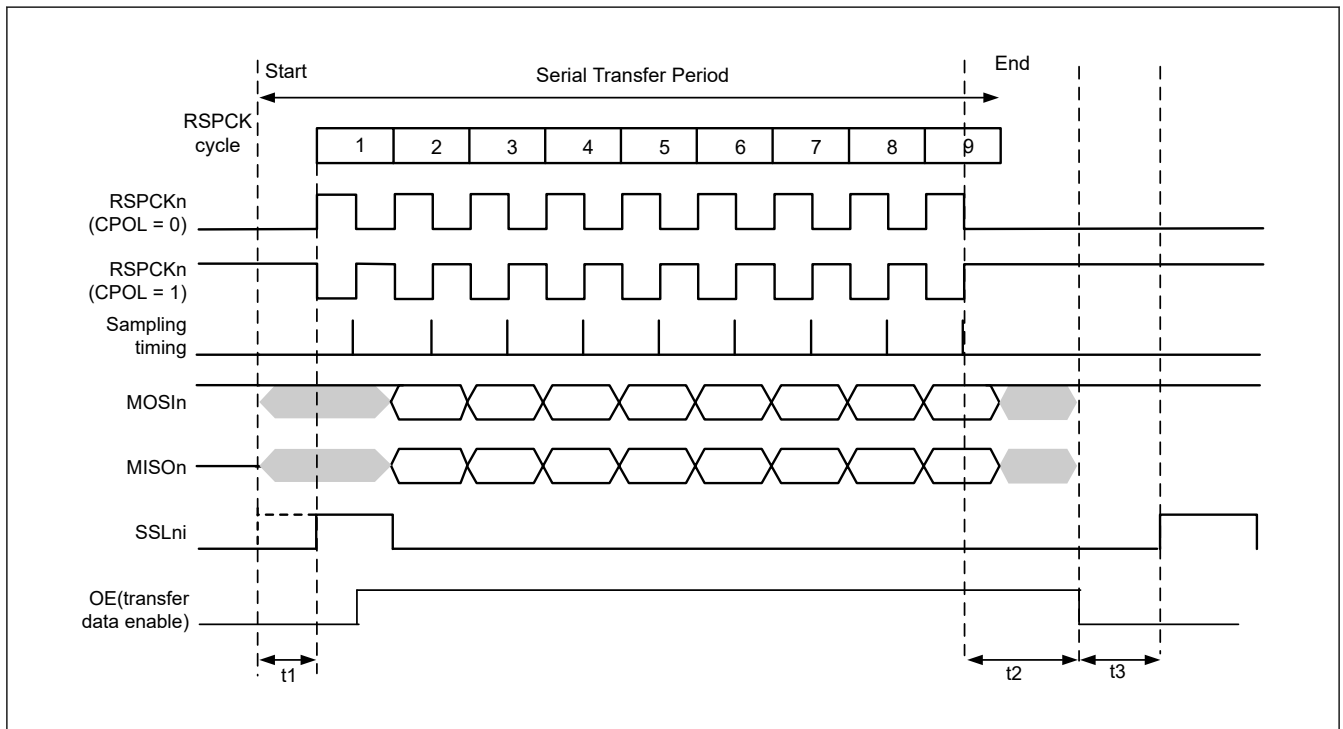
t1, t2, and t3 are the same as those when CPHA = 0. For a description of t1, t2, and t3 when the SPI of the MCU is in master mode, see section 34.3.12.1. Master mode operation.

[In the Motorola-SPI case]



**Figure 34.28 SPI transfer format when CPHA = 1, SPFRF = 0**

[In the TI-SSP case]



**Figure 34.29 SPI transfer format when CPHA = 1, SPFRF = 1**

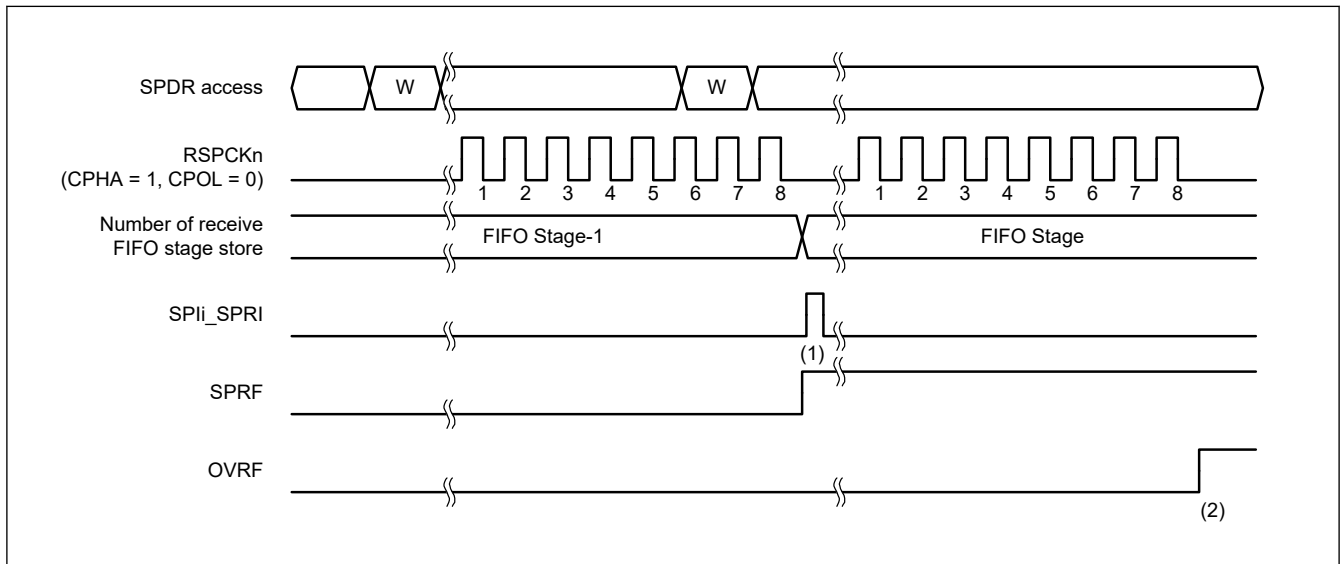
### 34.3.6 Communication Operating Mode

Transmit-Receive serial communication, transmit-only operation, and Receive-only operation are selected by setting the Communication Mode Select bits (TXMD [1:0]) of the SPI control register (SPCR).

SPDR access described in [Figure 34.30](#), [Figure 34.31](#), [Figure 34.32](#) shows an access to the SPI data register (SPDR). W shows a write cycle.

### 34.3.6.1 Transmit-Receive Serial Communication (TXMD[1:0] = 00b)

Figure 34.30 shows an example of operation when the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 00b. In the example, the SPI performs an 8-bit serial transfer when the SPDCR2.TTRG is 0, the SPDCR2.RTRG is FIFO stage - 1, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.



**Figure 34.30 Operation example when SPCR.TXMD[1:0] = 00b**

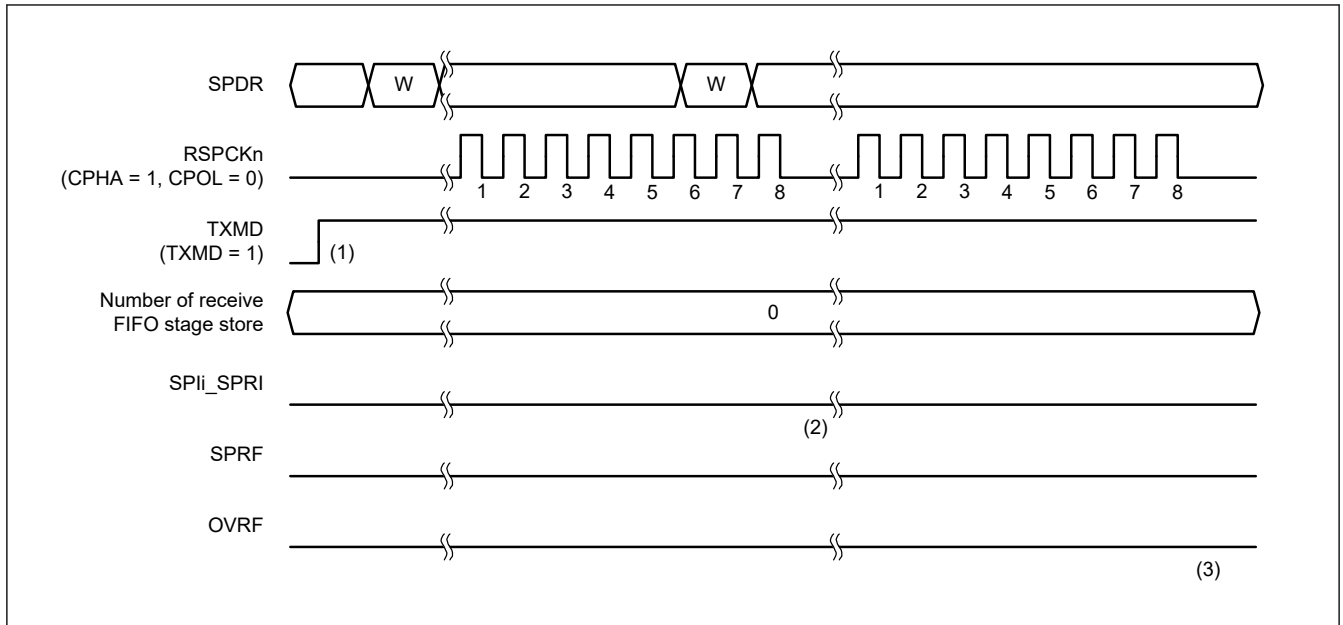
The operation of the flags at timings (1) and (2) in Figure 34.30 is as follows:

1. When serial transfer ends while the number of SPDR receive buffer store matches the number of frames set in SPDCR2.RTRG, the SPI generates a receive buffer full interrupt request (SPi\_SPRI), the SPI sets the SPSR.SPRF flag to 1, and copies the received data in the shift register to the receive buffer.
2. When serial transfer ends with data for the number of FIFO stages stored in the SPDR receive buffer, the SPI sets the SPSR.OVRF flag to 1, and discards the received data in the shift register. For details about the operation of the SPSR.OVRF flag, see [section 34.3.10.1. Overrun errors](#).

In Transmit-Receive serial communication (TXMD[1:0] = 00b), transmit data is transmitted and receive data is received. Therefore, the SPRF flag and the OVRF flag are set to 1 at timings (1) and (2) respectively.

### 34.3.6.2 Transmit-Only Serial Communications (TXMD[1:0] = 01b)

Figure 34.31 shows an example of operation when the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 01b. In this example, the SPI performs an 8-bit serial transfer when the SPDCR2.TTRG is 0, the SPDCR2.RTRG is 0, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.



**Figure 34.31 Operation example when SPCR.TXMD[1:0] = 01b**

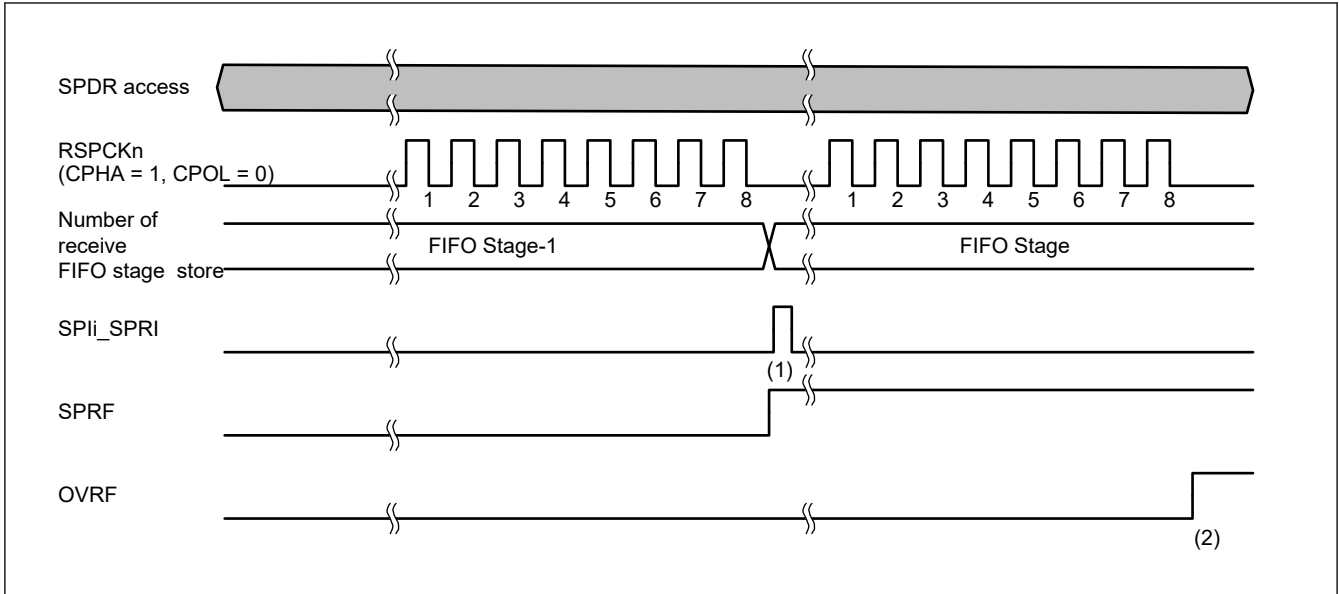
The operation of the flags at timings (1) to (3) in Figure 34.31 is as follows:

1. Make sure there is no data left in the receive buffer (the SPSR.SPRF flag is 0) and the SPSR.OVRF flag is 0 before entering transmit-only mode (SPCR.TXMD[1:0] = 01b).
2. When a serial transfer ends without receiving data in the receiving FIFO of SPDR, if the transmit-only mode is selected (SPCR.TXMD[1:0] = 01b), the SPSR.SPRF flag retains the value of 0, and the SPI does not copy the data in the shift register to the receive buffer.
3. Because the receive buffer of SPDR does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

In transmit-only mode (SPCR.TXMD[1:0] = 01b), the SPI transmits data but does not receive data. Therefore, the SPSR.SPRF and SPSR.OVRF flags remain 0 at timings (1) to (3).

### 34.3.6.3 Receive-Only Serial Communication (TXMD[1:0] = 10b)

Figure 34.32 shows an example of operation when the communication mode select bit (TXMD[1]) in the SPI control register (SPCR) is set to 1. In this example, the SPI performs an 8-bit serial transfer when the SPDCR2.TTRG is FIFO stage - 1, the SPDCR2.RTRG is 0, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.



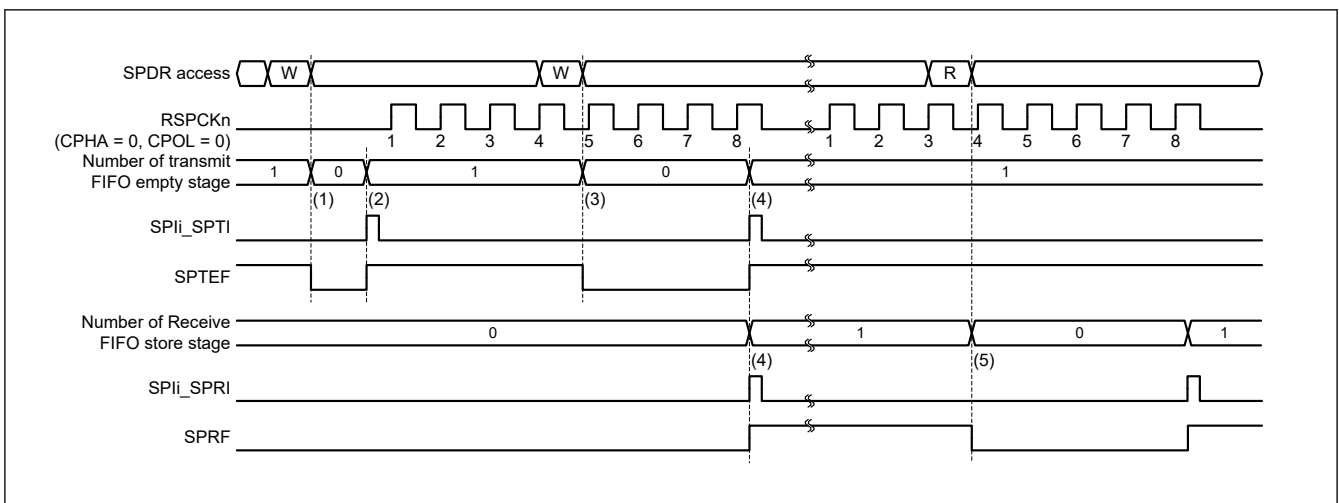
**Figure 34.32 Example of Operation when SPCR.TXMD[1:0] = 10b**

The following describes operation of flags at timings (1) and (2) in the figure above.

1. When serial transfer ends while the number of SPDR receive buffer store matches the number of frames set in SPDCR2.RTRG, the SPI generates a receive buffer full interrupt request (SPli\_SPRI), the SPI sets the SPSR.SPRF flag to 1, and copies the received data in the shift register to the receive buffer
2. When serial transfer ends with data for the number of FIFO stages stored in the SPDR receive buffer, the SPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

### 34.3.7 Transmit Buffer Empty and Receive Buffer Full Interrupts

Figure 34.33 show examples of operation of the transmit buffer empty interrupt (SPli\_SPTI (i = 0, 1)) and the receive buffer full interrupt (SPli\_SPRI). The SPDR register accesses shown in these figures indicate the conditions of access to the register, where W denotes a write cycle and R a read cycle. In Figure 34.33, the SPI performs an 8-bit serial transfer when SPCR.TXMD[1:0] bits are 00b, the SPDCR2.TTRG bit is 0, the SPDCR2.RTRG bit is 0, the SPCMDm.CPHA bit is 0, and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.



**Figure 34.33 Operation example of the SPli\_SPTI and SPli\_SPRI interrupts when CPHA = 0 and CPOL = 0 in master mode**

The operation of the SPI at timings (1) to (5) in Figure 34.33 is as follows:

1. When transmit data is written to SPDR with the transmit buffer of SPDR is before the next transfer data is set, the SPI writes data to the transmit buffer. When transmit data is written to SPDR in one processing routine using DTC / DMAC, the SPSR.SPTEF flag is cleared to 0 at the last access.
2. If the shift register is empty, the SPI copies the data in the transmit buffer to the shift register. At this time, if transmit FIFO empty stage number > TTRG value, then the SPI generates a transmit buffer empty interrupt request (SPI<sub>i</sub>\_SPTI), and sets the SPSR.SPTEF flag to 1. How a serial transfer is started depends on the SPI mode. For details, see [section 34.3. Operation](#), and [section 34.3.13. Clock Synchronous Operation](#).
3. When transmit data is written to SPDR either by the transmit buffer empty interrupt routine, or by the processing of the transmit buffer empty using the SPTEF flag, the SPI writes data to the transmit buffer. When the transmit data is written to SPDR in one processing routine using DTC / DMAC, the SPTEF flag is cleared to 0 at the last access. Because the data being transferred serially is stored in the shift register, the SPI does not copy the data in the transmit buffer to the shift register.
4. When the serial transfer ends with the receive buffer of SPDR > FIFO stage number, the SPI copies the receive data in the shift register to the receive buffer, generates a receive buffer full interrupt request (SPI<sub>i</sub>\_SPRI), and sets the SPRF flag to 1. Because the shift register becomes empty on completion of the serial transfer, if the next transfer data is set in the transmit FIFO before the serial transfer ended, the SPI sets the SPTEF flag to 1 and copies data in the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, on completion of the serial transfer, the SPI determines that the shift register is empty, so data transfer from the transmit buffer to the shift register is enabled.
5. When SPDR is read either by the receive buffer full interrupt routine or processing of the receive buffer full interrupt using the SPRF flag, the receive data can be read. If the received data is read from SPDR in one processing routine using DTC / DMAC, the SPRF flag is cleared to 0 at the last access.

When transmit data is written to the SPDR register while no empty stages in the transmit FIFO, the SPI does not update data in the transmit buffer. When writing to SPDR, always use either a transmit buffer empty interrupt request or check the empty or processing of the transmit buffer empty interrupt using the SPTEF flag. To use a transmit buffer empty interrupt, set the SPTIE bit in SPCR to 1. If the SPI function is disabled (the SPCR.SPE bit is 0), set the SPTIE bit to 0.

When serial transfer ends while data is stored in the receive FIFO for the number of FIFO stages, the SPI does not copy data from the shift register to the receive buffer, and it detects an overrun error (see [section 34.3.10. Error Detection](#)). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an SPI receive buffer full interrupt, set the SPCR.SPRIE bit to 1.

Transmission and reception interrupts or the associated IELSR<sub>n</sub>.IR flags (where n is the interrupt vector number) in the ICU can be used to confirm the states of the transmit and receive buffers.

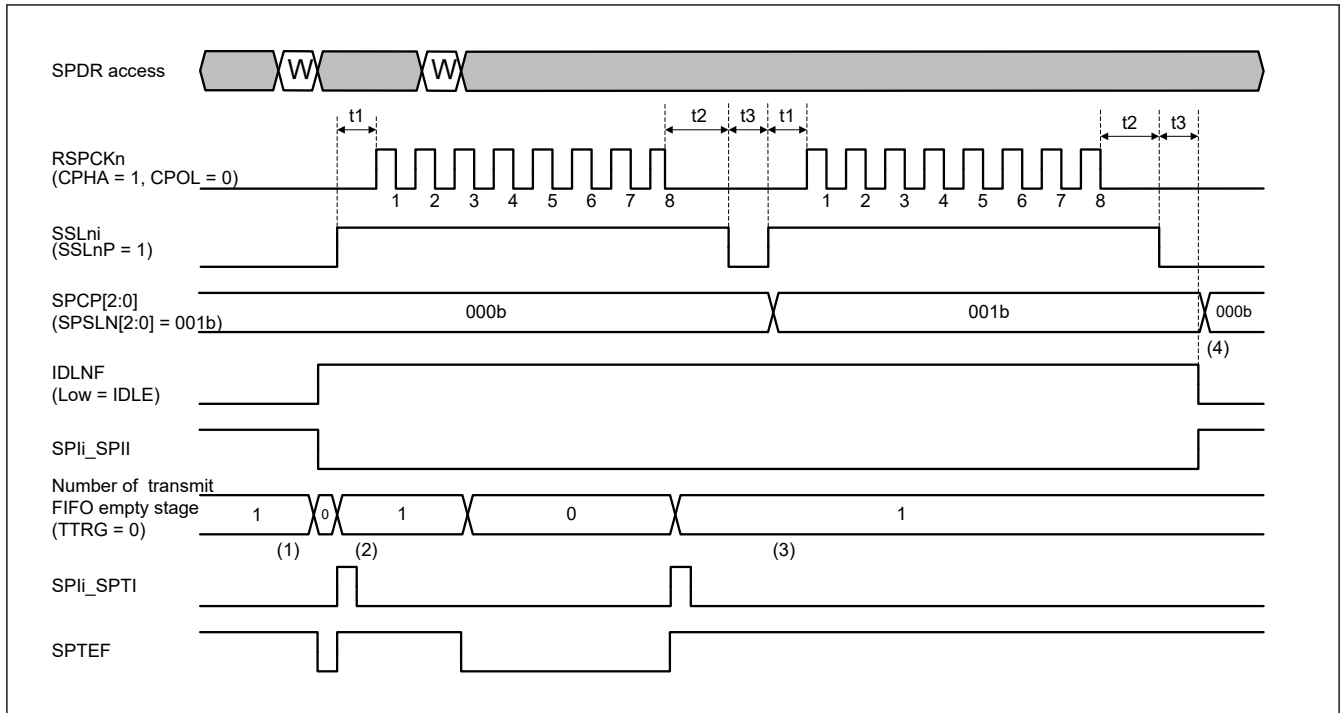
Similarly, the SPTEF and SPRF flags can be used to confirm the states of the transmit and receive buffers. See [section 13, Interrupt Controller Unit \(ICU\)](#) for the interrupt vector numbers.

### 34.3.8 Idle Interrupt

When the SPCP[2:0] of the SPI status register (SPSR) becomes 000b (start of sequence control), the IDLNF flag in the SPI status register (SPSR) is set to 1 and an idle interrupt request is made during master mode operation. An interrupt request is also made by clearing the SPCR.SPE bit to 0.

[In the Motorola-SPI case]

[Figure 34.34](#) shows an example of idle interrupt operation during normal operation.



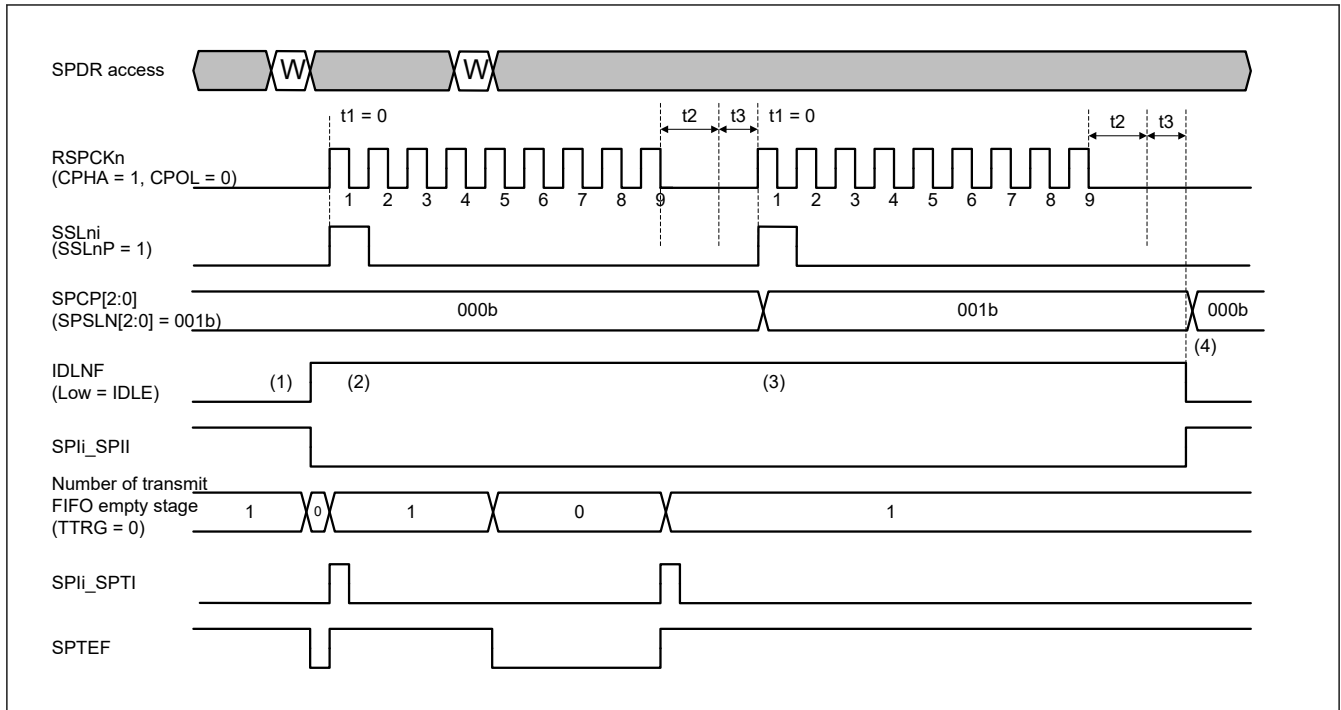
**Figure 34.34 Example of Idle Interrupt Operation (Master mode / Motorola-SPI)**

1. At the start of transmission, if the next transfer data is not set in the transmission buffer, the IDLNF flag is 0 (IDLE). Writing transmit data sets the IDLNF flag to 1 (BUSY). When the SPIIE bit in the SPI control register (SPCR) is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPIIE bit to 0 before starting transmission.
2. After transmission has started, the IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.
3. The SPCP[2:0] bits change the command to the next command at the end of t3 cycle. When the next command is not 000b, the IDLNF flag remains unchanged even when the next transmit data has not been written.
4. The IDLNF flag is cleared to 0 (IDLE) at the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPIIE bit is 1 currently, an SPIi\_SPII (i = 0, 1) interrupt is output.

[In the TI-SSP case]

Figure 34.35 shows an example of idle interrupt operation during normal operation.





**Figure 34.35 Example of Idle Interrupt Operation (Master mode / TI-SSP)**

1. At the start of transmission, if the next transfer data is not set in the transmission buffer, the IDLNF flag is 0 (IDLE). Writing transmit data makes sets the IDLNF flag to 1 (BUSY). When the SPIIE bit in the SPI control register (SPCR) is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPIIE bit to 0 before starting transmission.
2. After transmission has started, the IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.
3. The SPCP[2:0] bits change the command to the next command at the end of t3 cycle. When the next command is not 000b, the IDLNF flag remains unchanged even when the next transmit data has not been written.
4. The IDLNF flag is cleared to 0 (IDLE) at the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPIIE bit is 1 currently, an SPIi\_SPII interrupt is output.

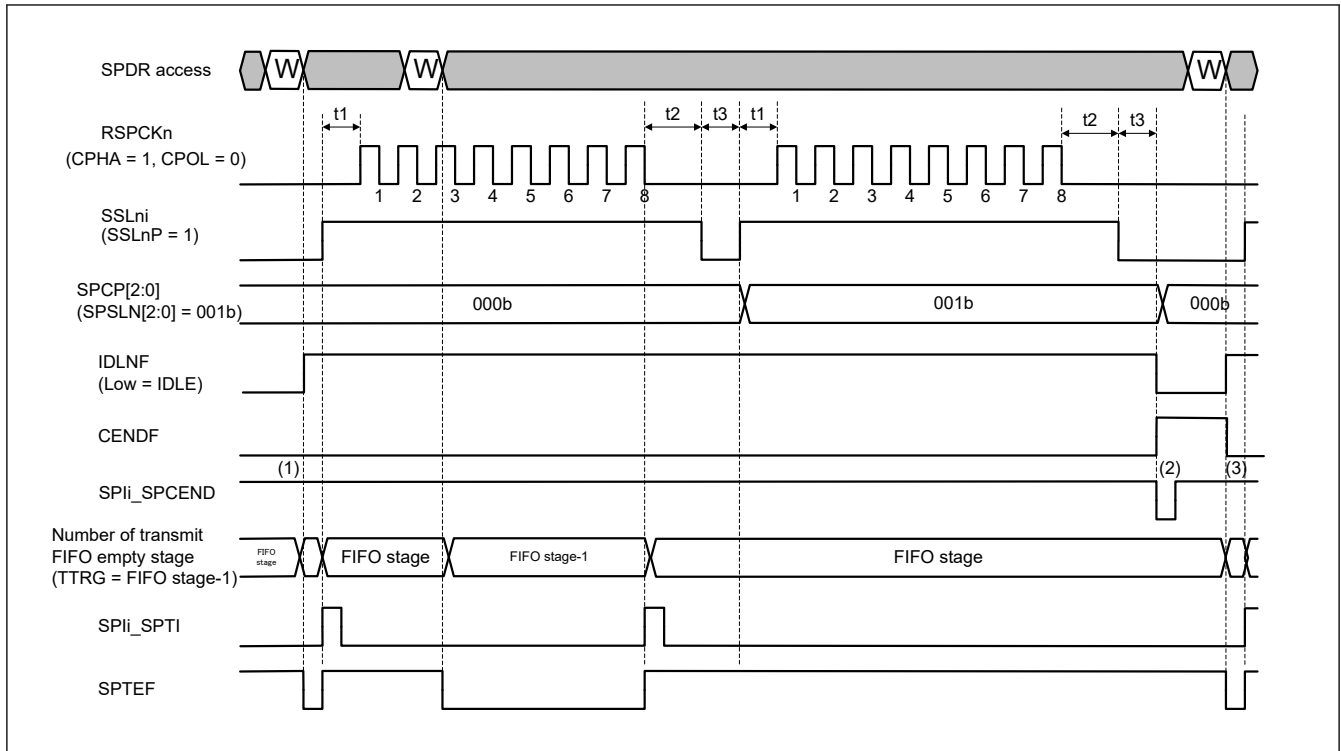
### 34.3.9 Communication End Interrupt

#### 34.3.9.1 Transmit-Receive/Transmit in Master Mode

See the description of the CENDF bit in [section 34.2.9. SPSR : SPI Status Register](#) for the setting / clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Master Mode.

[In the Motorola-SPI case]

[Figure 34.36](#) shows an example of communication end interrupt operation during transmit-receive/transmit master mode.

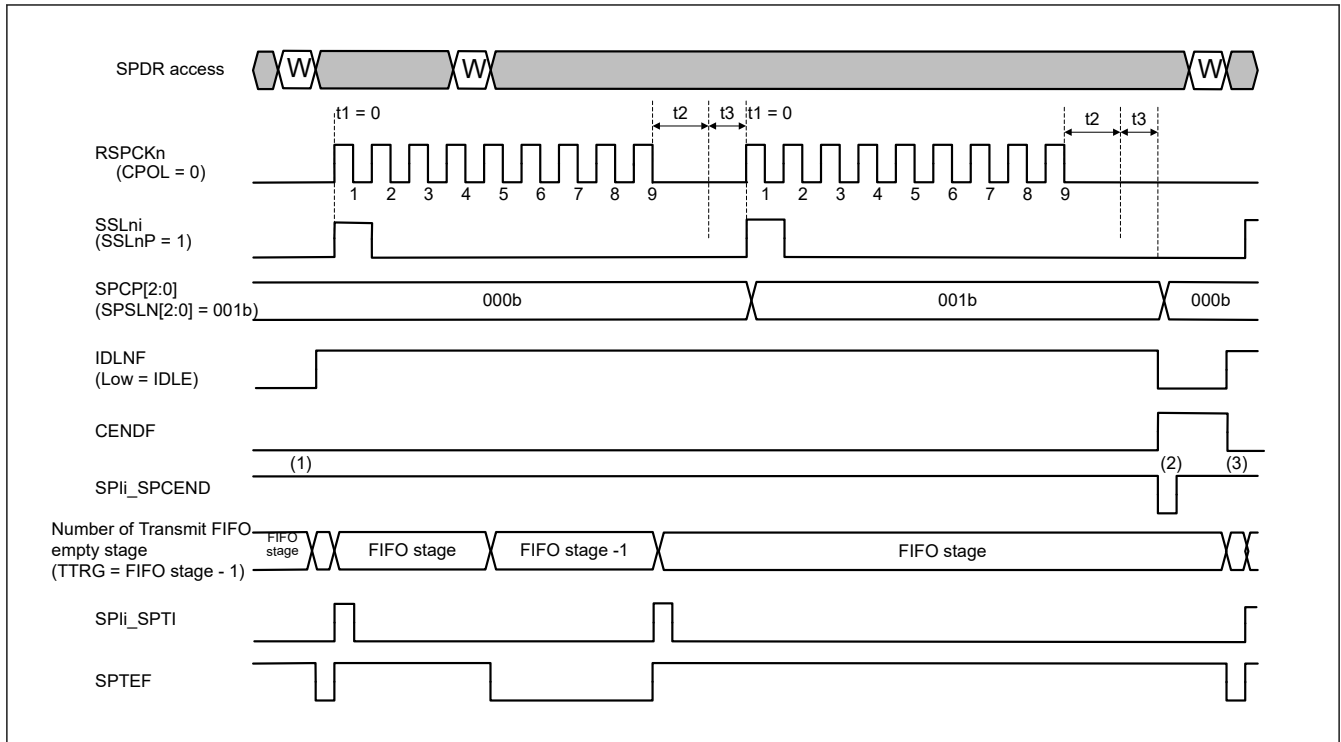


**Figure 34.36 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit Master mode/ Motorola-SPI)**

1. The CENDF flag is 0 and the level of SPIi\_SPCEND (i = 0, 1) is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle because the next command is 000b and there is no next transmit data, and then the SPIi\_SPCEND interrupt output when the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

[In the TI-SSP case]

Figure 34.37 shows an example of communication end interrupt operation during transmit-receive/transmit-only master mode.



**Figure 34.37 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit-only Master mode/TI-SSP)**

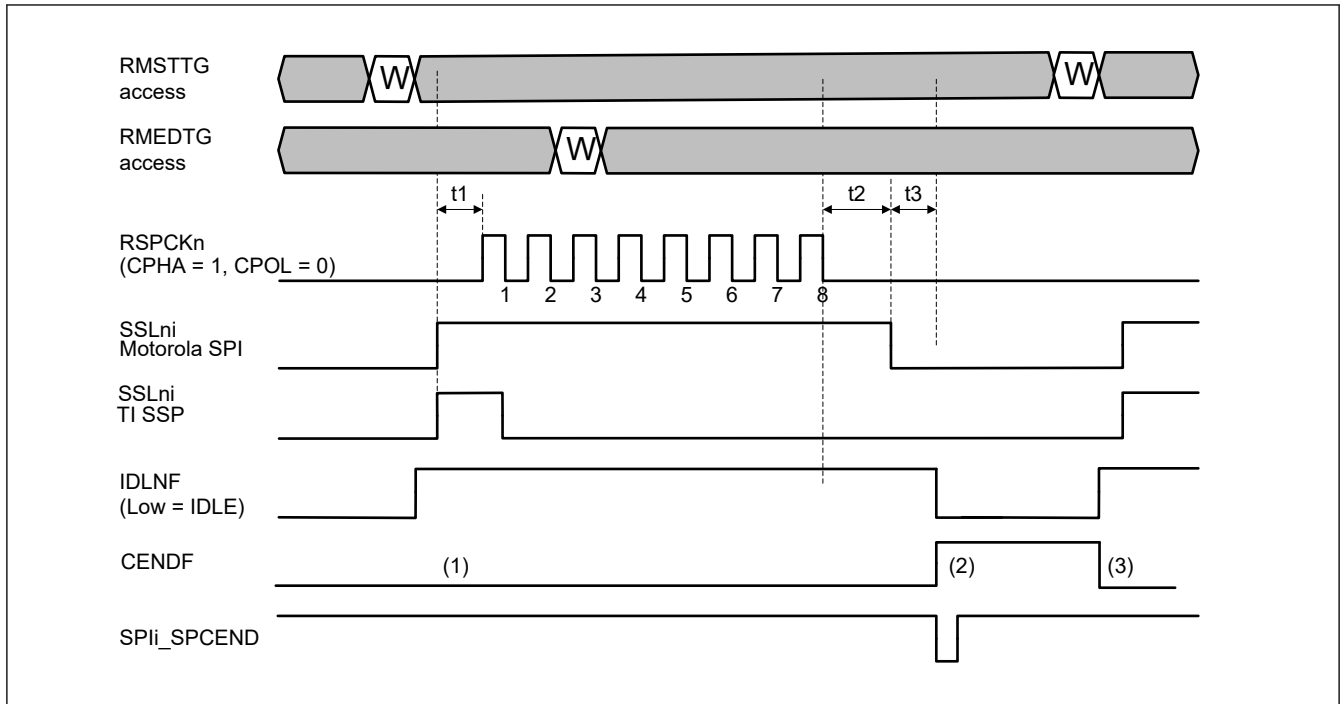
1. The CENDF flag is 0 and the level of SPIi\_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle because the next command is 000b and there is no next transmit data, and then the SPIi\_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

In slave mode operation, the output timing of the communication end interrupt is deferent due to the value of the SPCR.SPMS bit (SPI mode select bit), and the clear timing of the communication end interrupt is deferent due to the communication mode (transmit-receive or transmit-only or receive-only).

### 34.3.9.2 Receive-only in Master Mode

See the description of the CENDF bit in [section 34.2.9. SPSR : SPI Status Register](#) for the setting / clearing conditions of the communication completion flag during Receive-only in Master Mode.

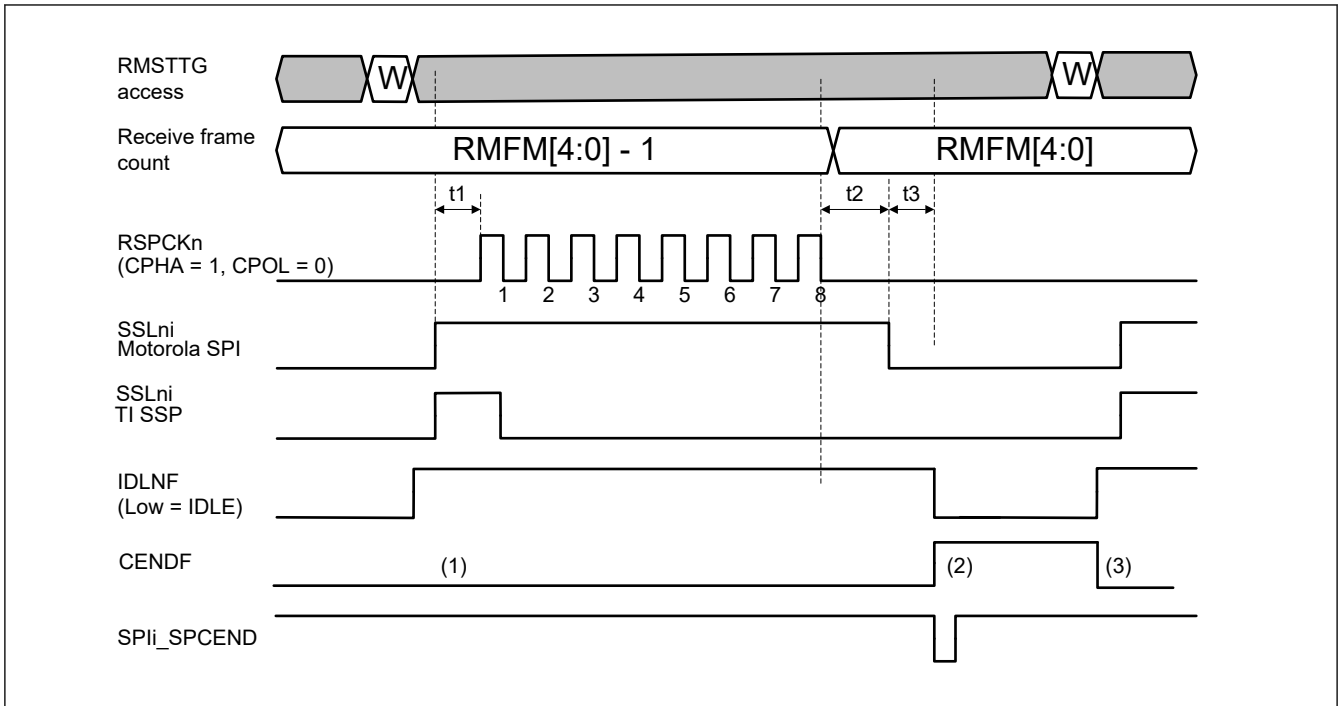
[Figure 34.38](#) shows an example of communication end interrupt operation during receive-only master mode at RMFM[4:0] = 0.



**Figure 34.38 Example of Communication End Interrupt Operation (Receive-only Master mode / Motorola-SPI) at RMFM [4:0] = 0**

1. The CENDF flag is 0 and the level of SPIi\_SPCEND is 1 before communication start. These have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle by writing 1 to RMEDTG during the communication frame. Then the SPIi\_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when writing 1 to RMSTTG. Also when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

Figure 34.39 shows an example of communication end interrupt operation during receive-only master mode at RMFM[4:0] ≠ 0.



**Figure 34.39 Example of Communication End Interrupt Operation (Receive-only Master mode / Motorola-SPI) at RMFM [4:0] ≠ 0**

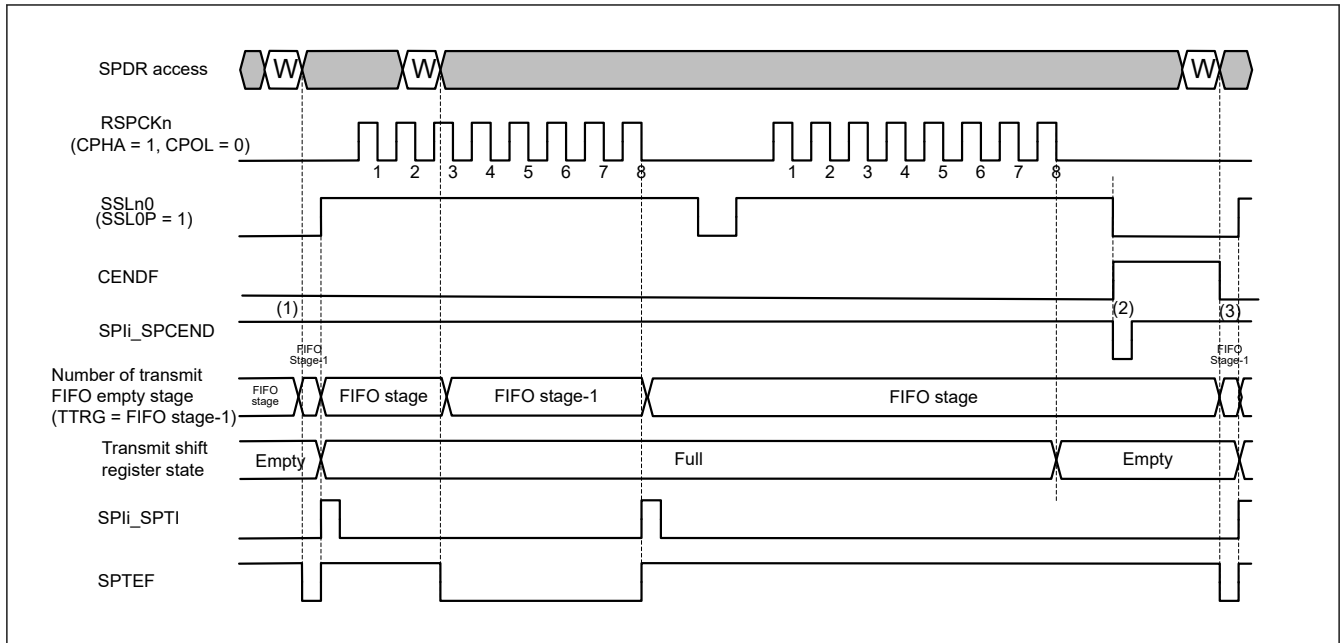
1. The CENDF flag is 0 and the level of SPli\_SPCEND is 1 before communication start. These have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the end of t3 cycle after receiving the number of frames set by RMFM[4:0]. Then the SPli\_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when writing 1 to RMSTTG. Also when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

### 34.3.9.3 Transmit-Receive/Transmit in Slave Mode on SPI Operation (4-wire)

See the description of the CENDF bit in [section 34.2.9. SPSR : SPI Status Register](#) for the setting / clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Slave Mode (4-wire).

[In the Motorola-SPI case]

[Figure 34.40](#) shows an example of communication end interrupt operation during transmit-receive/transmit slave mode on SPI operation.

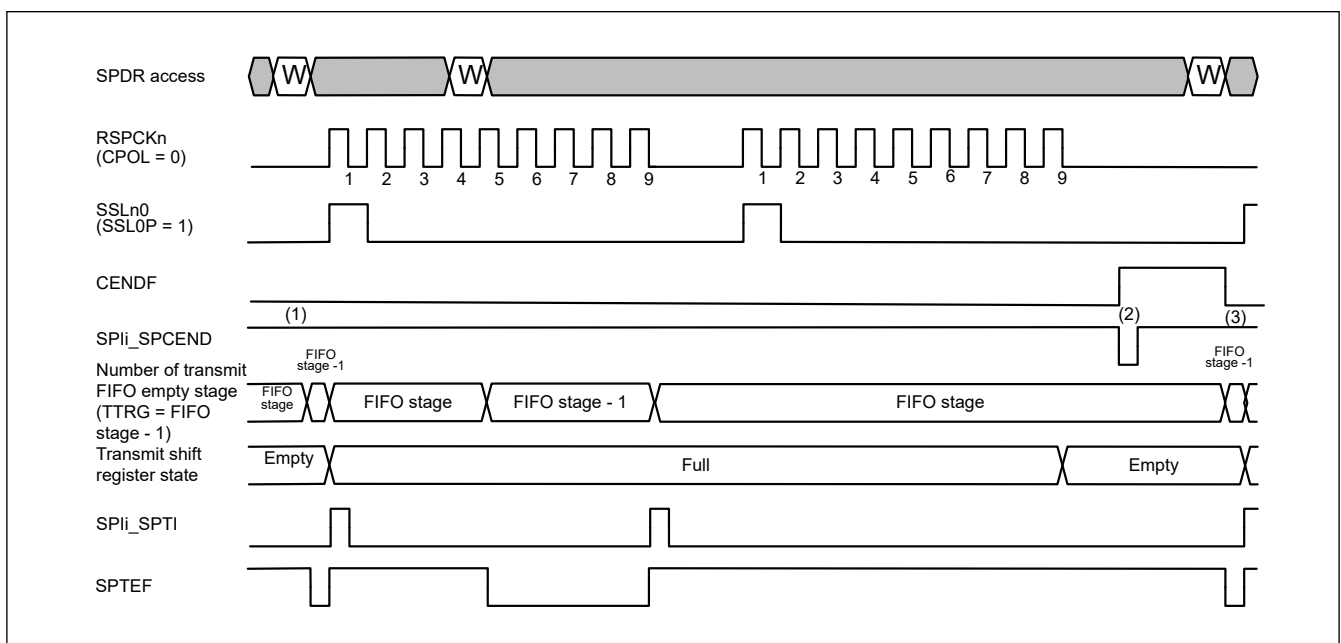


**Figure 34.40 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit Slave mode on SPI Operation/Motorola-SPI)**

1. The CENDF flag is 0 and the level of SPIi\_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the timing of SSLn0 negate when the next transfer data is not set in the transmit FIFO and the transmit shift register is empty. Then the SPIi\_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

[In the TI-SSP case]

Figure 34.41 shows an example of communication end interrupt operation during transmit-receive/transmit-only slave mode on SPI operation.



**Figure 34.41 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit-only Slave mode on SPI Operation / TI-SSP)**

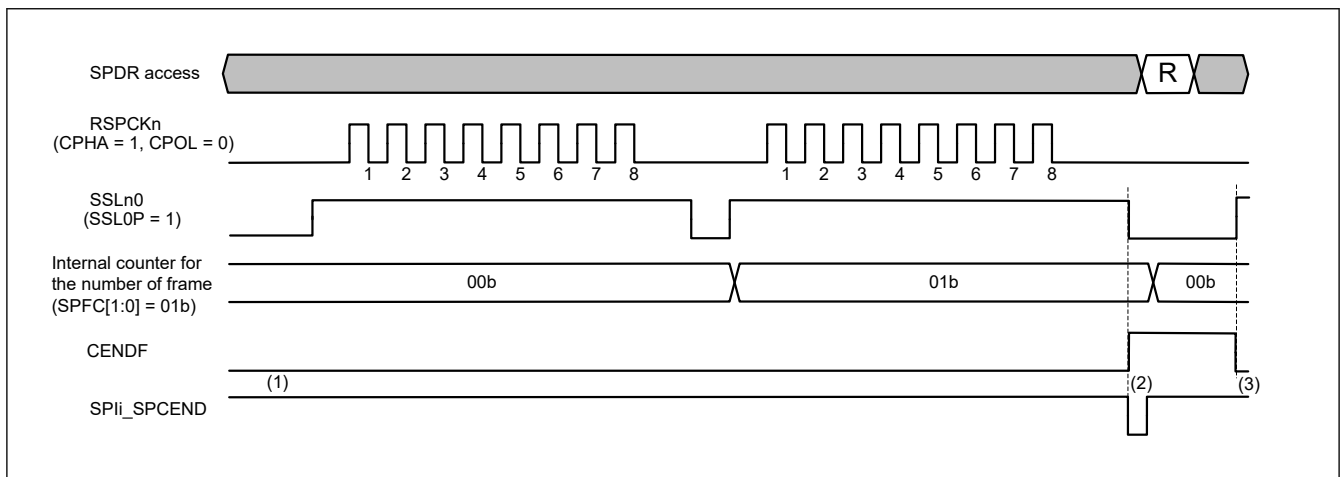
1. The CENDF flag is 0 and the level of SPIi\_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the RSPCKn last data bit sampling when the next transfer data is not set in the transmit FIFO and the transmit shift register is empty. Then the SPIi\_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX). Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

### 34.3.9.4 Receive Only in Slave Mode on SPI Operation (4-wire)

See the description of the CENDF bit in [section 34.2.9. SPSR : SPI Status Register](#) for the setting / clearing conditions of the communication completion flag during Receive-only in Slave Mode (4-wire).

[In the Motorola-SPI case]

[Figure 34.42](#) shows an example of communication end interrupt operation during receive only slave mode on SPI operation (4-wire).

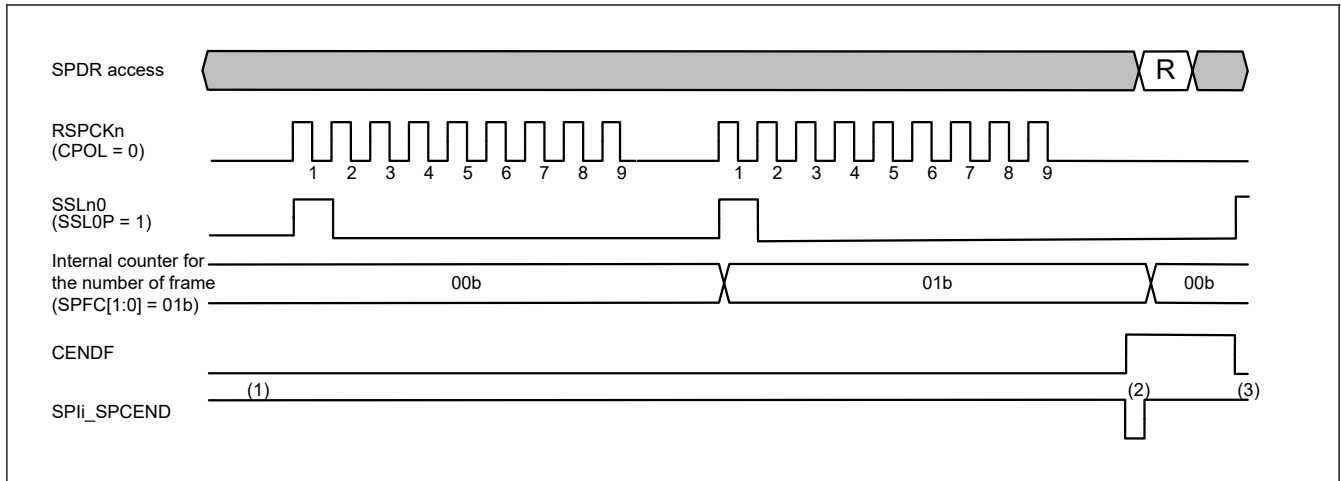


**Figure 34.42 Example of Communication End Interrupt Operation (Receive only Slave mode on SPI Operation / Motorola-SPI)**

1. The CENDF flag is 0 and the level of SPIi\_SPCEND is 1 before communication start, and these have kept during communication.
2. After the frames for SPFC set value in the SPI data control register (SPDCR) are stored in the receive buffer, the CENDF flag becomes 1 (communication completed) at the timing of SSLn0 negation. Then the SPIi\_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared at the SSLn0 assert when the next transmission start. Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

[In the TI-SSP case]

[Figure 34.43](#) shows an example of communication end interrupt operation during receive only slave mode on SPI operation (4-wire).



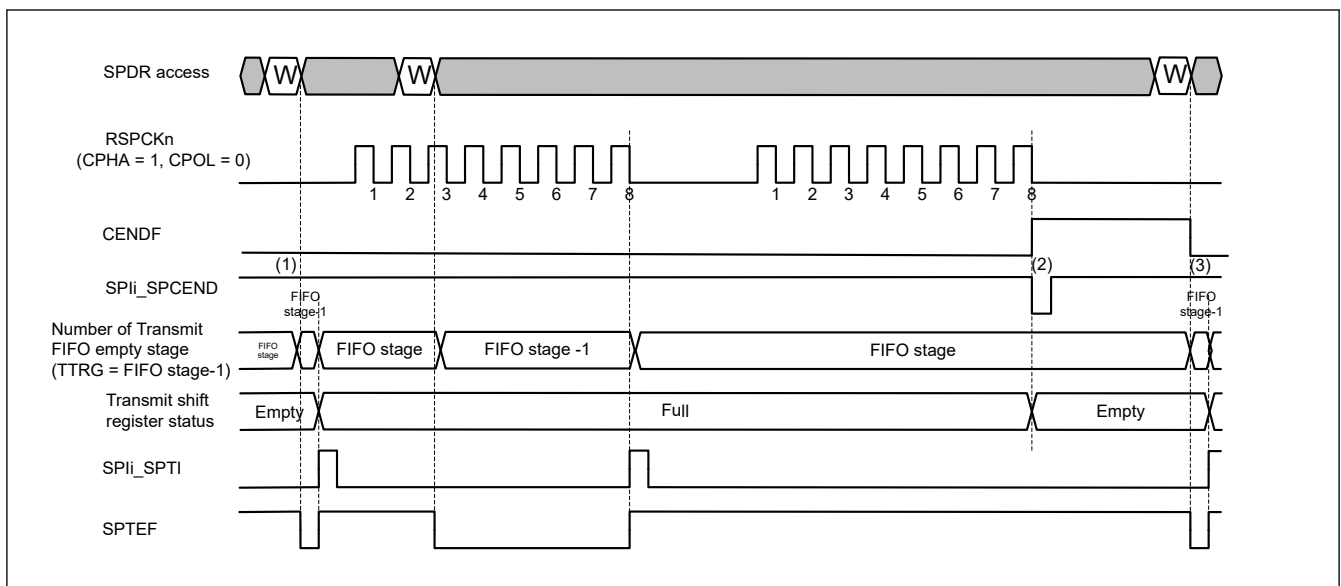
**Figure 34.43 Example of Communication End Interrupt Operation (Receive-only Slave mode on SPI Operation / TI-SSP)**

1. The CENDF flag is 0 and the level of SPIi\_SPCEND is 1 before communication start, and these have kept during communication.
2. The CENDF flag will be 1 (Communication End) at the RSPCK last data bit sampling when the last frame transmission ends. Then the SPIi\_SPCEND interrupt outputs with PCLK 1 cycle width if the CENDIE bit is 1.
3. The CENDF flag is cleared at the SSLn0 assert when the next transmission start. Or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

### 34.3.9.5 Transmit-Receive/Transmit in Slave Mode on Clock Synchronous Operation (3-wire)

See the description of the CENDF bit in [section 34.2.9. SPSR : SPI Status Register](#) for the setting / clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Slave Mode on Clock Synchronous (3-wire).

[Figure 34.44](#) shows an example of communication end interrupt operation during transmit-receive/transmit slave mode on clock synchronous operation (3-wire).



**Figure 34.44 Example of Communication End Interrupt Operation (Transmit-Receive/Transmit Slave mode on Clock Synchronous Operation)**

1. The CENDF flag is 0 and the level of SPIi\_SPCEND is 1 before communication start, and these have kept during communication.

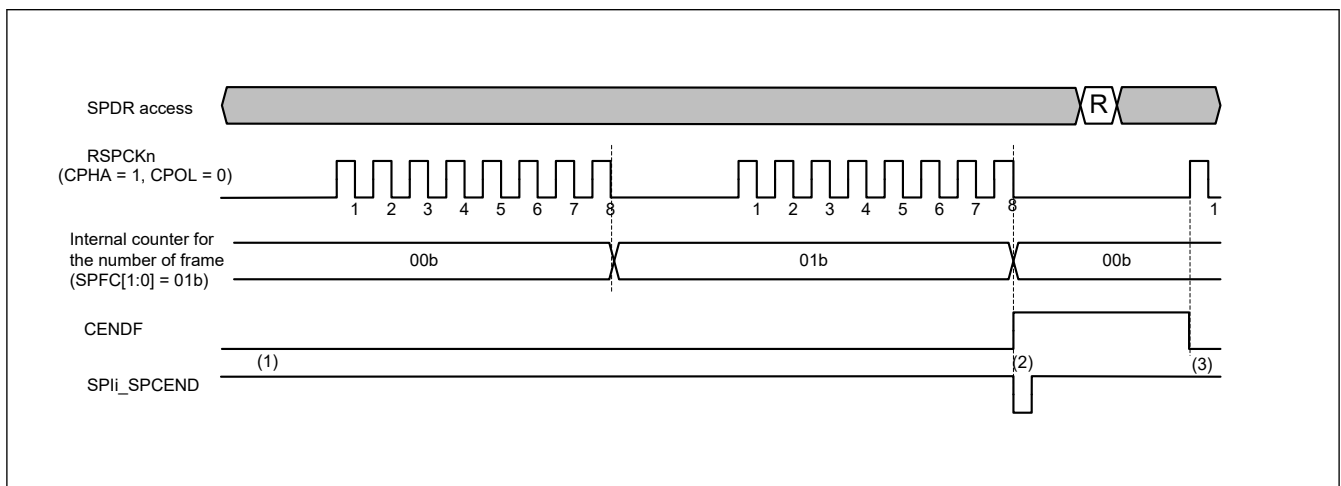


- When the next transfer data is not set in the transmit FIFO and the transmit shift register is empty, then the `SPIi_SPCEND` interrupt outputs with `PCLK` 1 cycle width if the `CENDIE` bit is 1.
- The `CENDF` flag is cleared when the next transmission data is written to the transmit buffer (`SPTX`). Or when 1 is written to the `SPSRC.CENDFC` bit, then the `CENDF` flag is 0.

### 34.3.9.6 Receive Only in Slave Mode on Clock Synchronous Operation (3-wire)

See the description of the `CENDF` bit in [section 34.2.9. SPSR : SPI Status Register](#) for the setting / clearing conditions of the communication completion flag during Receive -only in Slave Mode on Clock Synchronous (3-wire).

[Figure 34.45](#) shows an example of communication end interrupt operation during receive only slave mode on clock synchronous operation.

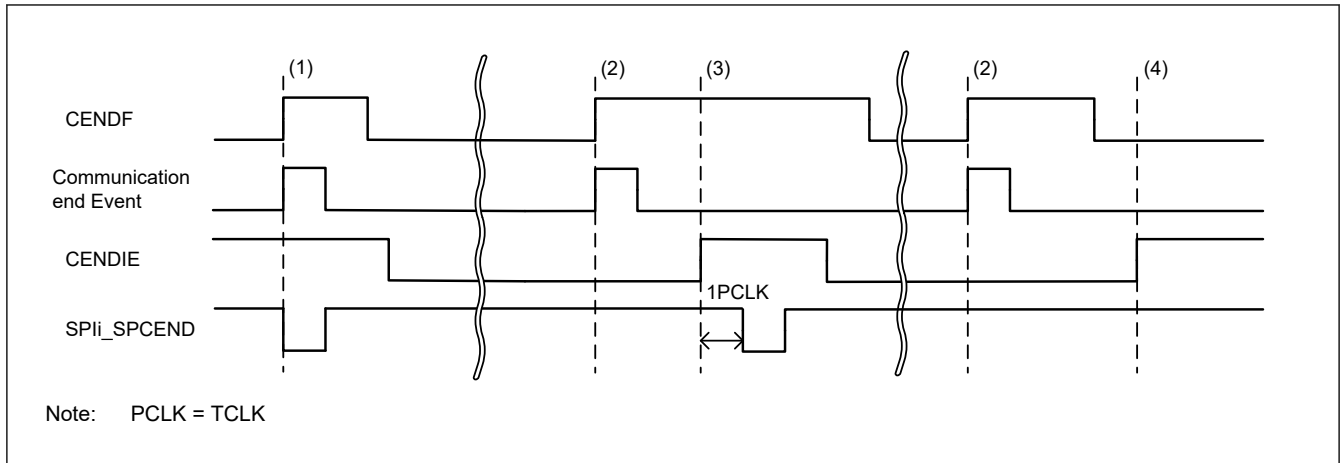


**Figure 34.45 Example of Communication End Interrupt Operation (Receive-only Slave mode on Clock Synchronous Operation)**

- The `CENDF` flag is 0 and the level of `SPIi_SPCEND` is 1 before communication start, and these have kept during communication.
- The `CENDF` flag is set to 1 (communication completed) at the timing of the last data bit sampling of `RSPCKn` in the last frame communication when the last frame of the SPI data control register (`SPDCR`) `SPFC` set value is received. Then the `SPIi_SPCEND` interrupt outputs with `PCLK` 1 cycle width if the `CENDIE` bit is 1.
- The `CENDF` flag is cleared at the first edge of `RSPCKn` for the next transmission. Or when 1 is written to the `SPSRC.CENDFC` bit, then the `CENDF` flag is 0.

### 34.3.9.7 Common Operation

In this chapter, the operation common to each mode / area option communication in [section 34.3.9.1. Transmit-Receive/ Transmit in Master Mode](#) to [section 34.3.9.6. Receive Only in Slave Mode on Clock Synchronous Operation \(3-wire\)](#) is explained. When the enable of SPI communication end interrupt (`CENDIE`) is 0, at the time of communication completion, a flag of communication end (`CENDF`) is set and an event of communication end is output, but no interrupt is output. However, if the enable of communication end interrupt (`CENDIE`) is set to 1 before clearing the flag of communication end (`CENDF`) while the enable of SPI function (`SPE`) is 1, the communication end interrupt (`SPIi_SPCEND`) is output.



**Figure 34.46 Example of Communication End Interrupt Operation (Enable control)**

1. When the enable of SPI communication end interrupt (CENDIE) is 1, at the time of communication completion, the following three are the same timing.
  - A flag of communication end (CENDF)
  - An event of communication end
  - The communication end interrupt (SPIi\_SPCEND)
2. When the enable of SPI communication end interrupt (CENDIE) is 0, at the time of communication completion, the following two are the same timing, but no interrupt.
  - A flag of communication end (CENDF)
  - An event of communication end
3. After (2), if the enable of communication end interrupt (CENDIE) is set when the enable of SPI function (SPE) and the flag of communication end (CENDF) are 1, the communication end interrupt (SPIi\_SPCEND) is output after 1 TCLK.
4. After (2), even if the enable of communication end interrupt (CENDIE) is set when the enable of SPI function (SPE) or the flag of communication end (CENDF) is 0, the communication end interrupt (SPIi\_SPCEND) is not output.

### 34.3.10 Error Detection

In normal SPI serial transfers, data written to the transmit buffer of SPDR is transmitted, and received data can be read from the receive buffer of SPDR. If access is made to SPDR, an abnormal transfer might occur, depending on the status of the transmit or receive buffer or the status of the SPI at the beginning or end of serial transfer.

If an abnormal transfer occurs, the SPI detects the event as an underrun error, overrun error, parity error, or mode fault error. [Table 34.9](#) lists the relationship between non-normal transfer operations and the SPI error detection function.

**Table 34.9 Relationship between non-normal transfer operations and SPI error detection (1 of 2)**

Operation	Occurrence condition	SPI operation	Error detection
1	SPDR is written while no empty stages in the transmit FIFO.	<ul style="list-style-type: none"> <li>• The contents of the transmit buffer are kept</li> <li>• Write data is missing</li> </ul>	None
2	SPDR is read while no data stored in receive FIFO.	The contents of the receive buffer and previously received data are output.	None
3	Serial transfer is started in slave mode when the SPI is not able to transmit data.	<ul style="list-style-type: none"> <li>• Serial transfer is suspended</li> <li>• Transmit or receive data is missing</li> <li>• Driving of the MISO<sub>n</sub> output signal is stopped</li> <li>• SPI function is disabled</li> </ul>	Underrun error
4	Serial transfer ends when data is stored in the receive FIFO for the number of FIFO stages.	<ul style="list-style-type: none"> <li>• Keeps the contents of the receive FIFO</li> <li>• Missing receive data</li> </ul>	Overrun error

**Table 34.9 Relationship between non-normal transfer operations and SPI error detection (2 of 2)**

Operation	Occurrence condition	SPI operation	Error detection
5	An incorrect parity bit is received during full-duplex synchronous serial communication with the parity function enabled in following mode: <ul style="list-style-type: none"> <li>• Transmit-receive master mode</li> <li>• Receive-only master mode</li> <li>• Transmit-receive slave mode</li> <li>• Receive-only slave mode</li> </ul>	The parity error flag is asserted	Parity error
6	The SSLn0 input signal is asserted when the serial transfer is idle in multi-master mode.	<ul style="list-style-type: none"> <li>• Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped</li> <li>• SPI function is disabled</li> </ul>	Mode fault error
7	The SSLn0 input signal is asserted during serial transfer in multi-master mode.	<ul style="list-style-type: none"> <li>• Serial transfer is suspended</li> <li>• Transmit or receive data is missing</li> <li>• Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped</li> <li>• SPI function is disabled</li> </ul>	Mode fault error
8	[In the Motorola-SPI case] The SSLn0 input signal is negated during serial transfer in slave mode.	<ul style="list-style-type: none"> <li>• Serial transfer is suspended</li> <li>• Transmit or receive data is missing</li> <li>• Driving of the MISO<sub>n</sub> output signal is stopped</li> <li>• SPI function is disabled</li> </ul>	Mode fault error
9	[In the TI-SSP case] The SSLn0 input signal is asserted during serial transfer in slave mode.	<ul style="list-style-type: none"> <li>• Serial transfer is suspended</li> <li>• Transmit or receive data is missing</li> <li>• Driving of the MISO<sub>n</sub> output signal is stopped</li> <li>• SPI function is disabled</li> </ul>	Mode fault error
10	After data is stored in the receive FIFO with SPDRES = 1, the number of stored data is less than the threshold value and no receive data is written for the set value of SPDRC [7:0]	Assert the receive data ready flag	Receive data ready

In operation 1 described in [Table 34.9](#), the SPI does not detect an error. To prevent data omission during writes to SPDR, the writes to SPDR must be executed using a transmit buffer empty interrupt request (when the SPSR.SPTEF flag is 1).

Similarly, the SPI does not detect an error in operation 2. To prevent extraneous data from being read, SPDR read must be executed with an SPI receive buffer full interrupt request (when the SPSR.SPRF flag is 1).

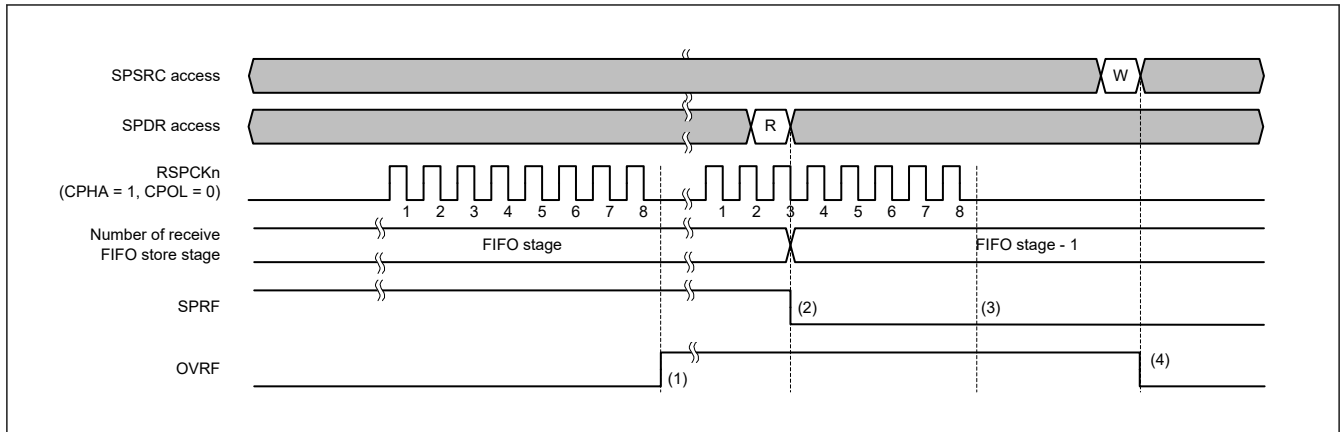
For information on the other errors, see the following sections:

- Underrun error, indicated in operation 3, see [section 34.3.10.4. Underrun errors](#)
- Overrun error, indicated in operation 4, see [section 34.3.10.1. Overrun errors](#)
- Parity error, indicated in operation 5, see [section 34.3.10.2. Parity errors](#)
- Mode fault error, indicated in operations 6 to 9, see [section 34.3.10.3. Mode fault errors](#)
- For the transmit and receive interrupts, see [section 34.3.7. Transmit Buffer Empty and Receive Buffer Full Interrupts](#).
- For the reception data ready in operations 10, see [section 34.3.10.5. Received data ready](#).

### 34.3.10.1 Overrun errors

If a serial transfer ends when the receive buffer of SPDR is full, the SPI detects an overrun error and sets the SPSR.OVRF flag to 1. When the OVRF flag is 1, the SPI does not copy data from the shift register to the receive buffer, so the data prior to the error occurrence is retained in the receive buffer. To set the OVRF flag to 0, issue a system reset or 1 is written to the SPSRC.OVRFC bit.

[Figure 34.47](#) shows an example of operation of the OVRF and SPRF flags. The SPSRC and SPDR accesses shown in [Figure 34.47](#) indicate the condition of accesses to the SPSRC and SPDR register, where W denotes a write cycle, and R a read cycle. In this example, the SPI performs an 8-bit serial transfer when SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCK<sub>n</sub> in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.



**Figure 34.47 Operation example of the OVRF and SPRF flags**

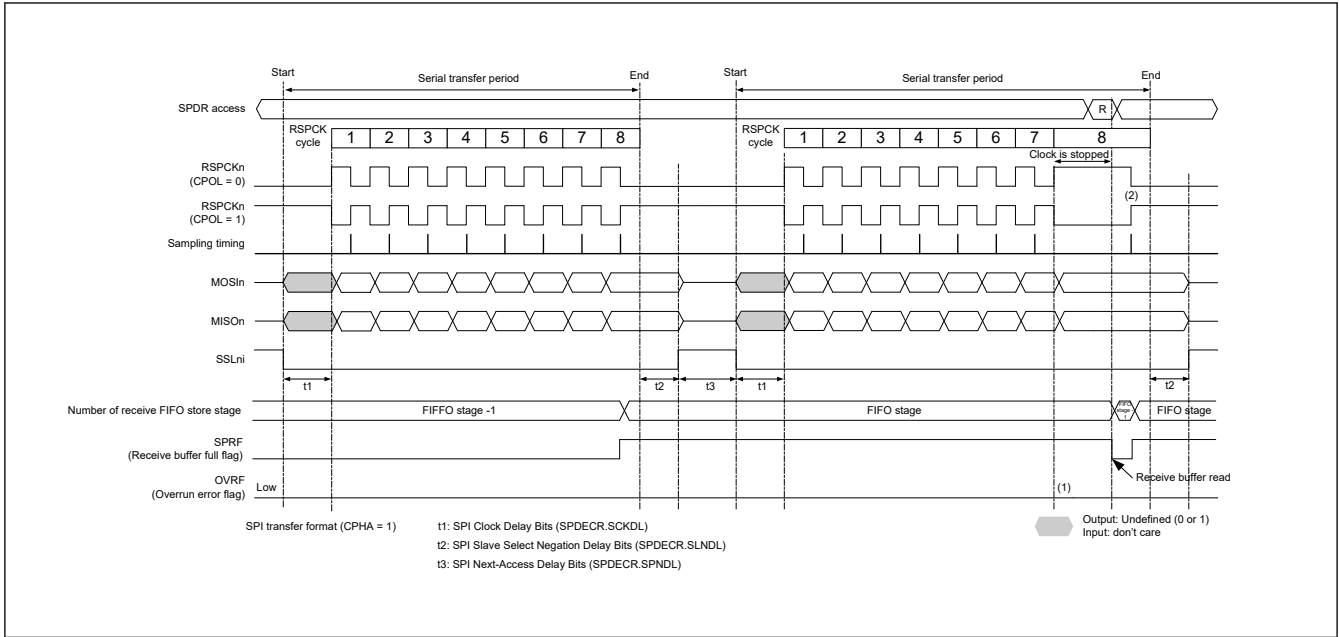
The operation of the flags at timings (1) to (4) in [Figure 34.47](#) is as follows:

1. When serial transfer ends while data is stored for the number of FIFO stages, the SPI detects an overrun error and sets the OVRF flag to 1. The SPI does not copy shift register data to the receive buffer. The SPI does not detect a parity error even when SPPE = 1. In master mode, the SPI copies the value of pointer to the SPI command register (SPCMDm) to the SPECMD[2:0] bits in the SPI status register (SPSR).
2. When SPDR is read, the SPI outputs the data in the receive buffer. At this time, the SPRF flag is cleared to 0 at the last access when the received data is read from SPDR in one processing routine using DTC / DMAC.
3. If the serial transfer ends with the OVRF flag set to 1 (overrun error occurred), the SPI does not copy data in the shift register to the receive buffer (the SPRF flag does not set to 1). A receive buffer full interrupt is not generated. Even when the SPPE bit is 1, parity errors are not detected. In an overrun error state when the SPI does not copy the received data from the shift register to the receive buffer, on termination of the serial transfer, the SPI determines that the shift register is empty. This enables data transfer from the transmit buffer to the shift register.
4. When 1 is written to the SPSRC.OVRFC bit, the SPSR.OVRF flag is cleared.

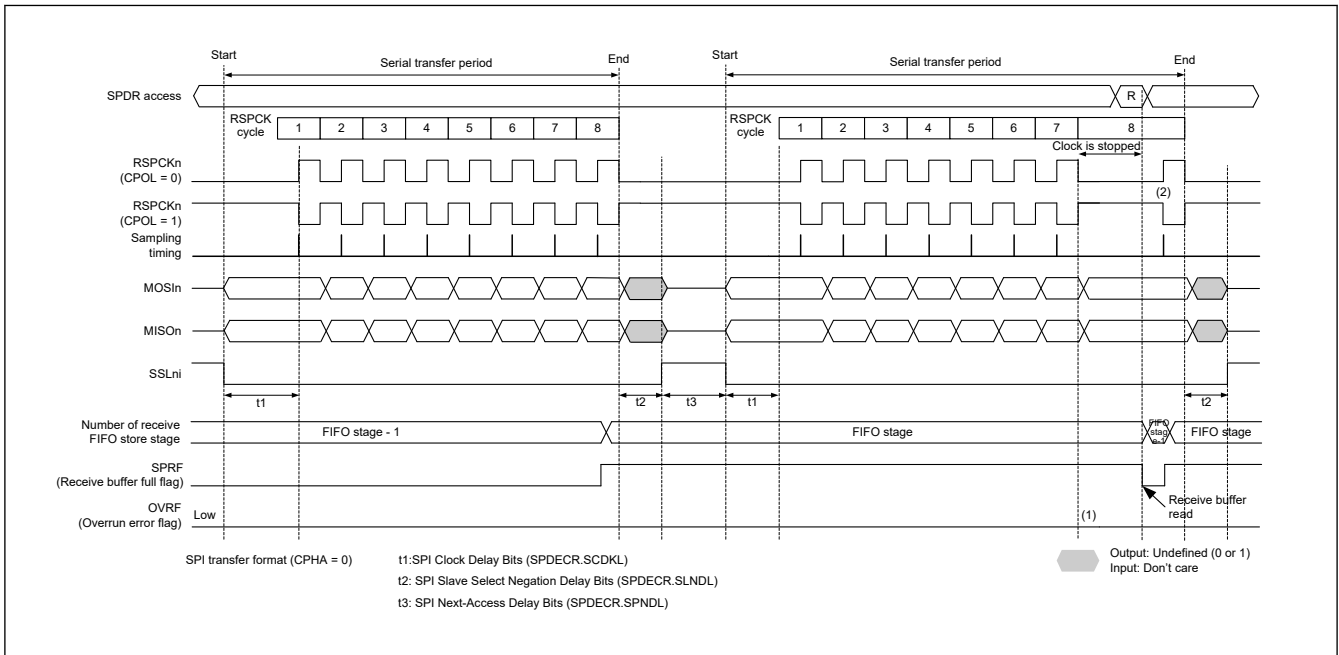
The occurrence of an overrun can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. When executing a serial transfer, you must ensure that overrun errors are detected early, for example by reading SPSR immediately after SPDR is read.

If an overrun error occurs and the OVRF flag sets to 1, normal reception operations cannot be performed until the OVRF flag is set to 0.

When the RSPCK auto-stop function is enabled (SPCR.SCKASE = 1) in master mode, an overrun error does not occur. [Figure 34.48](#) and [Figure 34.49](#) show the clock stop waveform when a serial transfer continues while the receive buffer is full in master mode.



**Figure 34.48 Clock stop waveform when serial transfer continues with data is stored for the number of FIFO stages in master mode (CPHA = 1)**

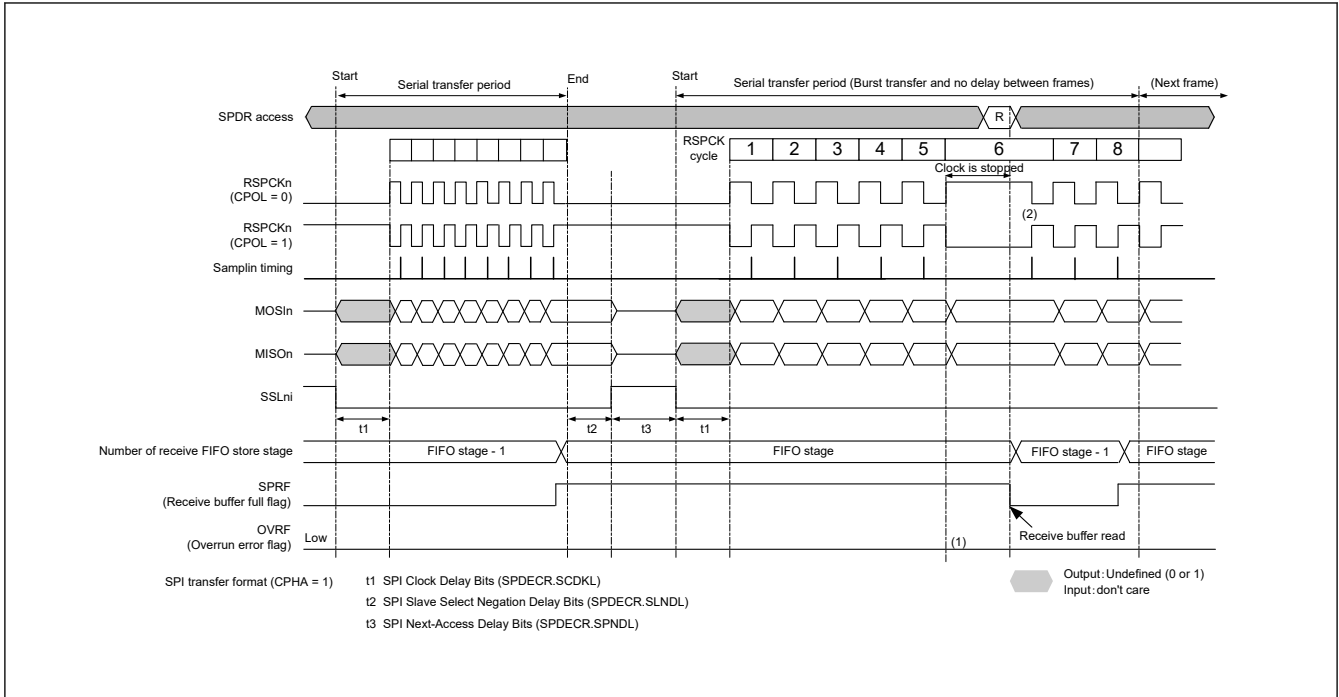


**Figure 34.49 Clock stop waveform when serial transfer continues with data is stored for the number of FIFO stages in master mode (CPHA = 0)**

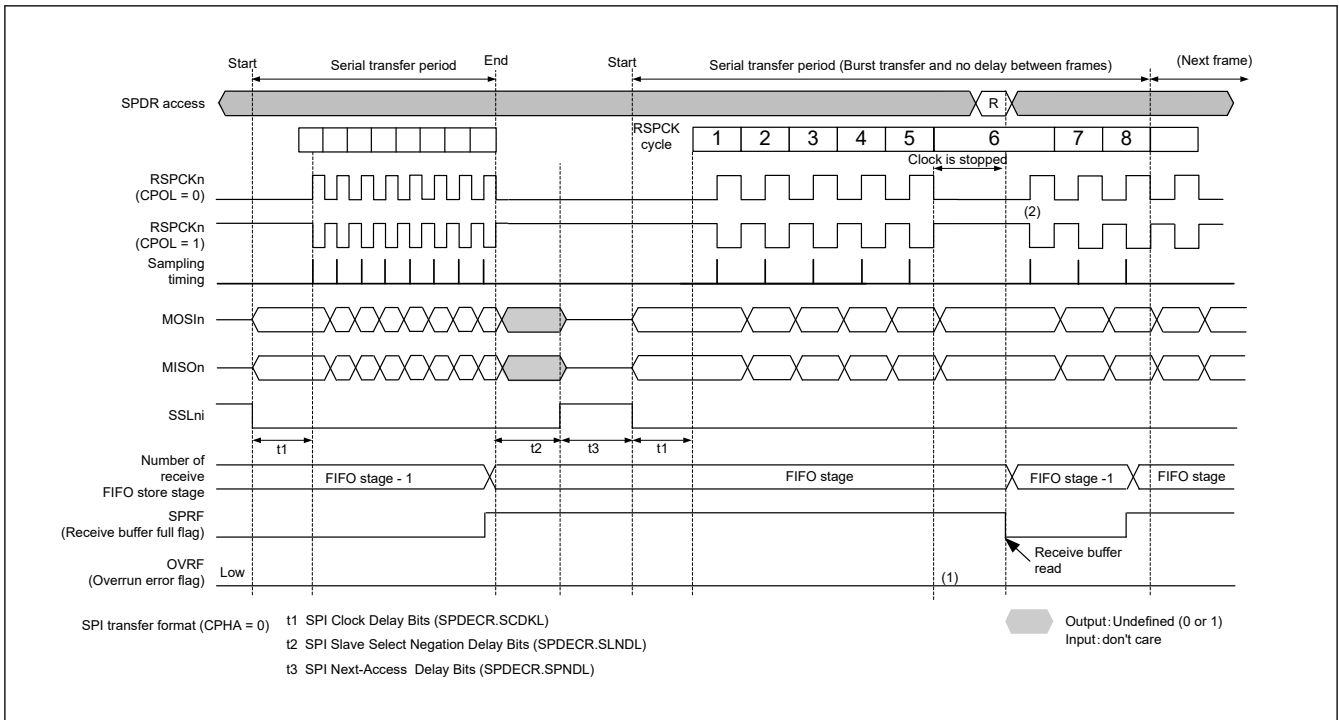
The operation of the flags at timings (1) and (2) in Figure 34.48 and Figure 34.49 is as follows:

1. While data is stored in the receive FIFO for the number of FIFO stages, the RSPCK clock is deactivated and no overrun error occurs.
2. If SPDR is read while the clock is stopped, data in the receive buffer can be read. The RSPCK clock restarts.

Overrun error does not occur when RSPCK automatic stop function is enabled for transfer with no delay of between frames during burst transfer in master mode. Figure 34.50 and Figure 34.51 show the clock stop waveform, when there is no delay between frames at burst transfer and the serial transfer continues in the data is stored in the receive FIFO for the number of FIFO stages.



**Figure 34.50 Clock Stop Waveform when Serial Transfer Continues in the Receive Buffer Full with data is stored for the number of FIFO stages (at burst transfer and no delay between frames CPHA = 1)**



**Figure 34.51 Clock Stop Waveform when Serial Transfer Continues with data is stored for the number of FIFO stages in Master Mode (at burst transfer and no delay between frames CPHA = 0)**

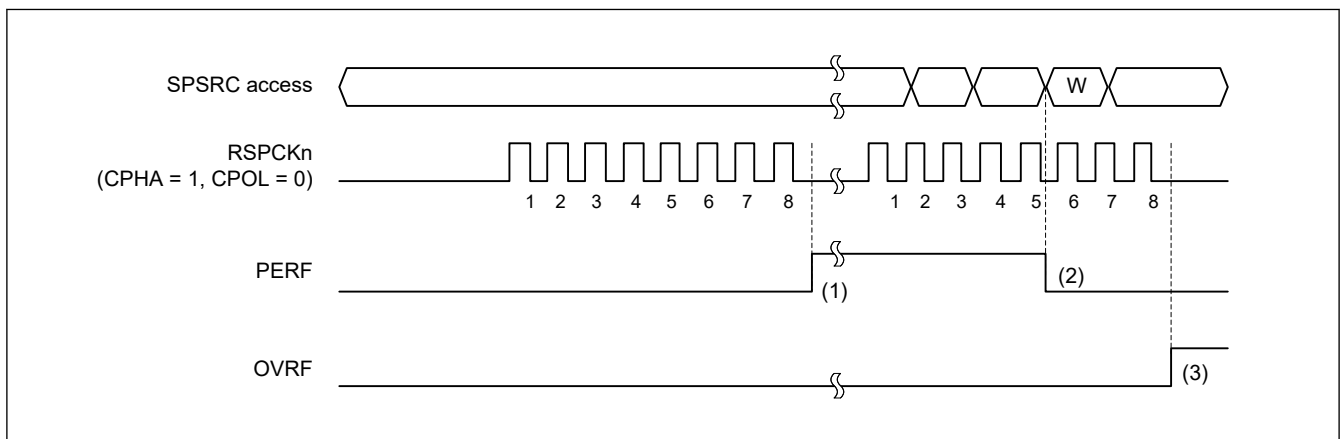
The following describes operation of flags at timings (1) and (2) in the figure above.

1. While the data is stored for the number of FIFO stages, the RSPCK clock is deactivated and no overrun error occurs.
2. Receive buffer data can be read by reading SPDR during clock stop. After the receive buffer data has been read the RSPCK clock restarts.

### 34.3.10.2 Parity errors

After transfer in transmit-receive or receive-only master mode, transmit-receive slave mode or receive only slave mode while the SPPE bit in the SPI control register (SPCR) is 1, the SPI checks occurrence of a parity error. When the SPI detects a parity error in received data, the PERF flag in the SPI status register (SPSR) is set to 1. While the OVRF flag is 1, the SPI does not copy shift register data to the receive buffer. Therefore, parity error in received data is not detected. To clear the PERF flag in SPSR to 0, issue a system reset or 1 is written to the SPSRC.PERFC bit.

Figure 34.52 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 34.52 indicates the condition of access to the register, where W denotes a write cycle, and R a read cycle. In this example, full-duplex serial communication is performed while the SPCR.SPPE bit is 1. The SPI performs an 8-bit serial transfer when SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given for RSPCKn in the waveform represent the number of RSPCK cycles, such as the number of transferred bits.



**Figure 34.52 Operation example of the OVRF and PERF flags**

The operation of the flags at timings (1) to (3) in Figure 34.52 is as follows:

1. When the SPI does not detect an overrun error and terminates the serial transfer, the SPI copies shift register data to the receive buffer. When the SPI checks the received data and detects a parity error at this time, the PERF flag is set to 1. In master mode, the SPI copies the value of pointer to the SPI command register (SPCMDm) to the SPECm[2:0] bits in the SPI Data control register 2 (SPDCR2).
2. When 1 is written to the SPSRC.PERFC bit, then clear the PERF flag.
3. When the SPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The SPI does not perform parity error detection at this time.

Parity errors can be checked for by either reading the SPSR register or using an SPI error interrupt and reading the SPSR register. When executing a serial transfer, such checks are required to ensure early detection of parity errors. When the SPI is used in master mode, the pointer value to the SPCMDm register at the occurrence of the error can be checked by reading the SPDCR2.SPECM[2:0] bits.

### 34.3.10.3 Mode fault errors

The SPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1.

If the active level is input for the SSLn0 input signal of the SPI in multi-master mode, the SPI detects a mode fault error regardless of the status of the serial transfer, and sets the SPSR.MODF flag to 1.

On detecting the mode fault error, the SPI copies the value of the pointer to SPCMD to the SPECm[2:0] bits.

The active level of the SSLn0 signal is determined by the SPCR3.SSL0P bit.

When the MSTR bit is 0, the SPI operates in slave mode.

When the SPCR.MODFEN bit = 1 and the SPMS bit = 0 in slave mode, if the SSLn0 input signal is negated during the serial transfer period (from valid data drive start to final valid data latch), the SPI detects a mode fault error, while any of the following 2 conditions is met.

[In the Motorola-SPI case]

When the SSLn0 input signal is negated while serial data transfer.

[In the TI-SSP case]

When the SSLn0 input signal is asserted while serial data transfer. However, during burst transfer, no error is detected even if the SSLn0 input signal is asserted during the last bit of frame.

When the SPI detects a mode fault error, it stops driving output signals and clears the SPE bit in the SPCR register. When the SPE bit is cleared, the SPI function is disabled (as described in [section 34.3.12. SPI Operation](#)). In a multi-master configuration, the mastership can be released by stopping driving output signals and disabling the SPI function by using a mode fault error.

Whether a mode fault error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR. To detect a mode fault error without using an SPI error interrupt, poll SPSR. When the SPI is used in master mode, the pointer value to SPCMD when an error is present can be checked by reading the SPECMD[2:0] bits in SPSR.

While the MODF flag = 1, the SPI ignores writing 1 to the SPE bit. To enable the SPI function after a mode fault error is detected, clear the MODF flag to 0 without fail.

### 34.3.10.4 Underrun errors

While the SPI is operating in slave mode (SPCR.MSTR bit = 0) and the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 00b or 01b, if serial transfer is started before transmit data output is ready with the SPCR.SPE bit set to 1 (SPI function enabled), the SPI detects an underrun error and sets the SPSR.MODF and SPSR.UDRF flags to 1.

On detecting an underrun error, the SPI stops the driving of output signals and clears the SPCR.SPE bit to 0 (see [section 34.3.11. Initializing the SPI](#)).

The occurrence of underrun errors can be checked either by reading SPSR or by using an SPI error interrupt and reading SPSR. Detecting underrun errors without using the SPI error interrupt requires polling of SPSR.

When the MODF flag is 1, writing 1 to the SPE bit is ignored by the SPI. To enable the SPI function after the detection of an underrun error, the MODF flag must be set to 0.

### 34.3.10.5 Received data ready

When SPCR.TXMD[1:0] = 00b, 01b, or 11b, and SPCR2.SPDRC[7:0] ≠ 0x00, after receiving data in the receive FIFO during communication (SPE = 1), SPSR.SPDRF flag is set to 1 when the received data is not stored even after the number of received FIFOs is equal to or less than ≤ the threshold value and the value set in SPDRC[7:0] has elapsed.

When the receive data ready is detected, the interrupt and event link output can be selected as SPIi\_SPRI or SPIi\_SPEI with the SPCR.SPDRF bit.

Figure 34.53 shows an example of reception data ready detection operation.

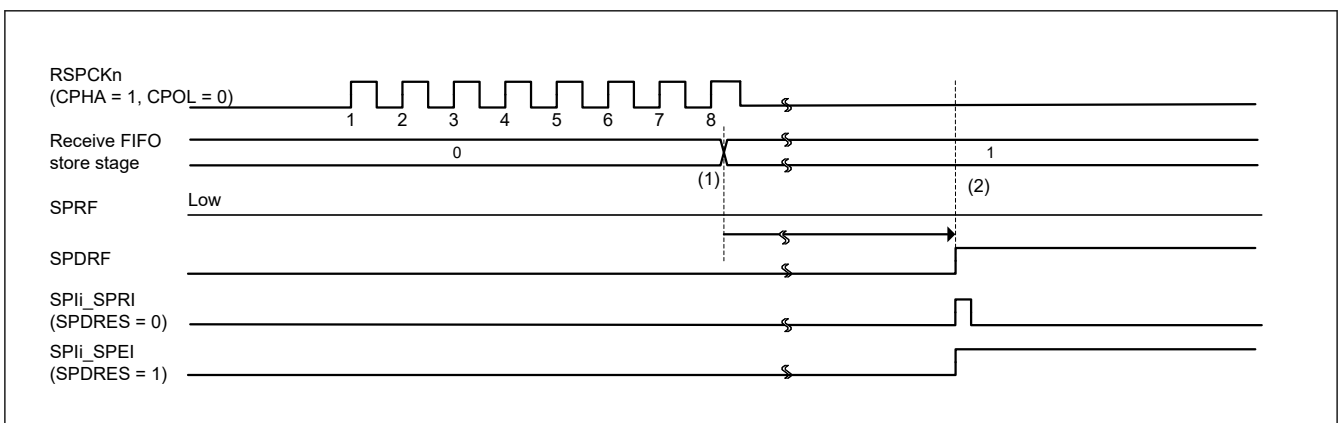


Figure 34.53 Received data ready

The following describes the operation at the timings indicated by (1) and (2) in the figure.



(1) Store the received data in the receive FIFO. SPRF is 0, because receive FIFO store stage  $\leq$  number of frames set by SPDCR2.RTRG[1:0].

(2) Set SPDRF and assert SPI<sub>i</sub>\_SPRI or SPI<sub>i</sub>\_SPEI because there is no writing to the receive FIFO for the amount of SPDRFC [7:0] set from above (1).

### 34.3.11 Initializing the SPI

If 0 is written to the SPCR.SPE bit or if the SPI sets the SPE bit to 0 because it detected a mode fault error or an underrun error, the SPI disables the SPI function and initializes some of the module functions. When a system reset is generated, the SPI initializes all of the module functions. This section describes initialization by clearing of the SPCR.SPE bit and by a system reset.

#### 34.3.11.1 Initialization by clearing of the SPCR.SPE bit

When the SPCR.SPE bit is set to 0, the SPI initializes by:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the SPI
- Initializing the transmit buffer of the SPI (the SPSR.SPTEF flag sets to 1)

Initialization by clearing of the SPE bit does not initialize the control bits of the SPI. For this reason, the SPI can be started in the same transfer mode in use prior to initialization when the SPE bit is set to 1 again.

The SPSR.CENDF, SPSR.SPRF, SPSR.OVRF, SPSR.MODF, SPSR.PERF, and SPSR.UDRF flags are not initialized, and the value of the SPDCR2.SPECM[2:0] and SPDCR2.SPCP[2:0] bits are not initialized. Therefore, even after the SPI is initialized, data from the receive buffer can be read to check the communication completion status and the error status during an SPI transfer.

The transmit buffer is initialized to an empty state (the SPSR.SPTEF flag sets to 1). Therefore, if the SPCR.SPTIE bit is set to 1 after SPI initialization, a transmit buffer empty interrupt is generated. To disable any transmit buffer empty interrupts when the SPI is initialized, write 0 to the SPTIE bit simultaneously while writing 0 to the SPE bit.

#### 34.3.11.2 Initialization by system reset

A system reset completely initializes the SPI by initializing all SPI control bits, status bits, and data registers, in addition to the requirements described in [section 34.3.11.1. Initialization by clearing of the SPCR.SPE bit](#).

### 34.3.12 SPI Operation

#### 34.3.12.1 Master mode operation

The only difference between single- and multi-master mode operation is the use of mode fault error detection (see [section 34.3.10. Error Detection](#)). In single-master mode, the SPI does not detect mode fault errors whereas in multi-master mode, it does. This section explains operations that are common to both modes.

##### (1) Starting a serial transfer

When data is written to the SPI data register (SPDR) while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTX<sub>n</sub>, n = 0 to 3) data in SPDR. When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the SPI copies data from the transmit buffer to the shift register and starts serial transfer. On copying transmit data to the shift register, the SPI changes the status of the shift register to full. On termination of the serial transfer, it changes the status of the shift register to empty. The status of the shift register cannot be referenced.

The polarity of the SSL<sub>n</sub> output pins depends on the SPCR3.SSL<sub>n</sub>P (n = 0 to 3) bits settings. For details on the SPI transfer format, see [section 34.3.5. Transfer Formats](#).

##### (2) Terminating a serial transfer

[Except Receive-only in Master Mode]

After the SPI detects the RSPCK<sub>n</sub> edge corresponding to the final sampling timing regardless of the CPHA bit value in the SPI command register (SPCMD), the SPI terminates serial transfer. When the number of data stored in the receive FIFO is less than the number of FIFO stages, data is copied from the shift register to the receive buffer in the SPI data register (SPDR) after serial transfer.

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMD<sub>m</sub>.SPB[4:0] bit settings. The polarity of the SSL<sub>n</sub>i output pin depends on the SPCR3.SSL<sub>n</sub>P (n = 0 to 3) bits settings. For details on the SPI transfer format, see [section 34.3.5. Transfer Formats](#).

[Receive-only in Master Mode]

When any of the following 2 conditions is met, then SPI terminating the serial transfer.

- After the SPI detects the RSPCK<sub>n</sub> edge corresponding to the final sampling timing regardless of the CPHA bit value in the SPI command register (SPCMD), the SPI terminates serial transfer.
- When writing SPCR2.RMEDTG = 1 during the serial transfer period, SPI terminating the serial transfer.

When the number of data stored in the receive FIFO is less than the number of FIFO stages, data is copied from the shift register to the receive buffer in the SPI data register (SPDR) after serial transfer.

The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in master mode depends on the set value of the SPB[4:0] bits in the SPI command register (SPCMD). The SSL<sub>n</sub>i output signal polarity depends on the set SPI SSL<sub>i</sub> signal polarity bit (SPCR3.SSL<sub>i</sub>P) (i = 0 to 3) value. For details about the SPI transfer format, see [section 34.3.5. Transfer Formats](#).

### (3) Sequence control

The transfer format in master mode is determined as follows.

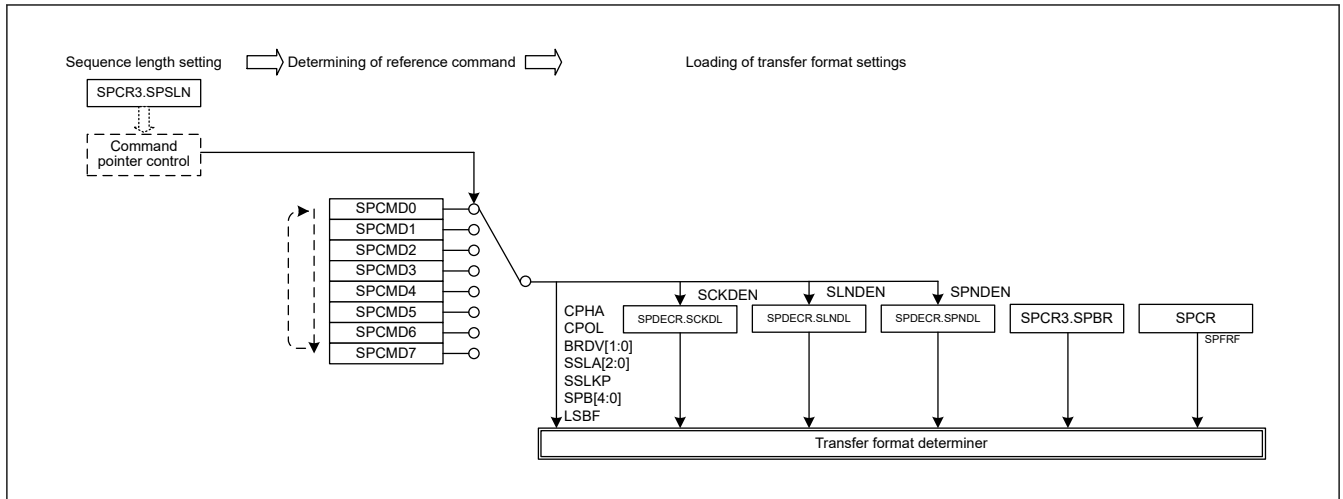
The transfer format used in master mode is determined by the SPCR3, SPCMD<sub>m</sub>, and SPDECR registers.

The SPCR3.SPSSLN[2:0] bits determine the sequence configuration for serial transfers that are executed by the SPI in master mode. The following items are set in the SPCMD<sub>m</sub> register:

- SSL<sub>n</sub>i pin output signal value
- MSB- or LSB-first
- Data length
- Some of the bit rate settings
- RSPCK<sub>n</sub> polarity and phase
- Whether SPDECR.SCKDL is to be referenced
- Whether SPDECR.SLNDL is to be referenced
- Whether SPDECR.SPNDL is to be referenced

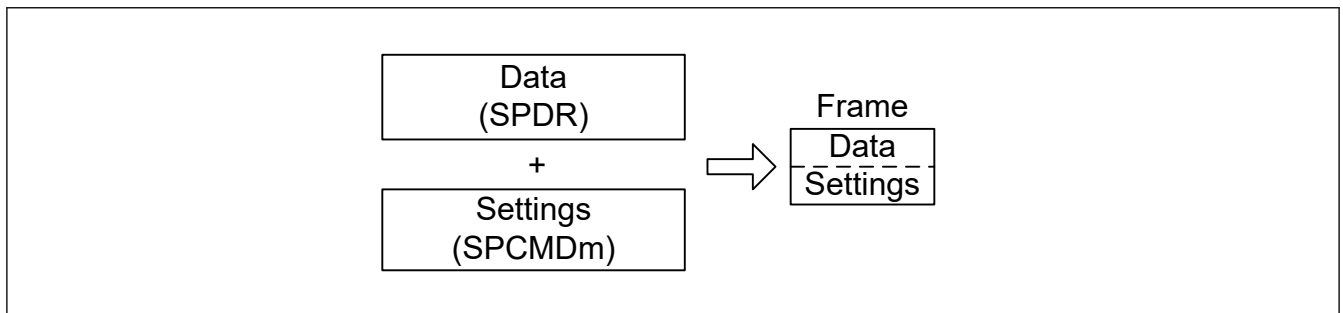
SPCR3.SPBR holds some of the bit rate settings, including SPDECR.SCKDL (SPI clock delay), SPDECR.SLNDL (SSL negation delay), and SPDECR.SPNDL (next-access delay).

Based on the sequence length assigned in SPCR3.SPSSLN, the SPI makes up a sequence comprised of a part or all of the SPCMD<sub>m</sub> register. The SPI contains a pointer to the SPCMD<sub>m</sub> register that makes up the sequence. The value of this pointer can be checked by reading the SPDCR2.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD<sub>0</sub>, and incorporates the SPCMD<sub>0</sub> settings into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command in the sequence, the SPI sets the pointer to SPCMD<sub>0</sub>, and in this way the sequence is executed repeatedly.



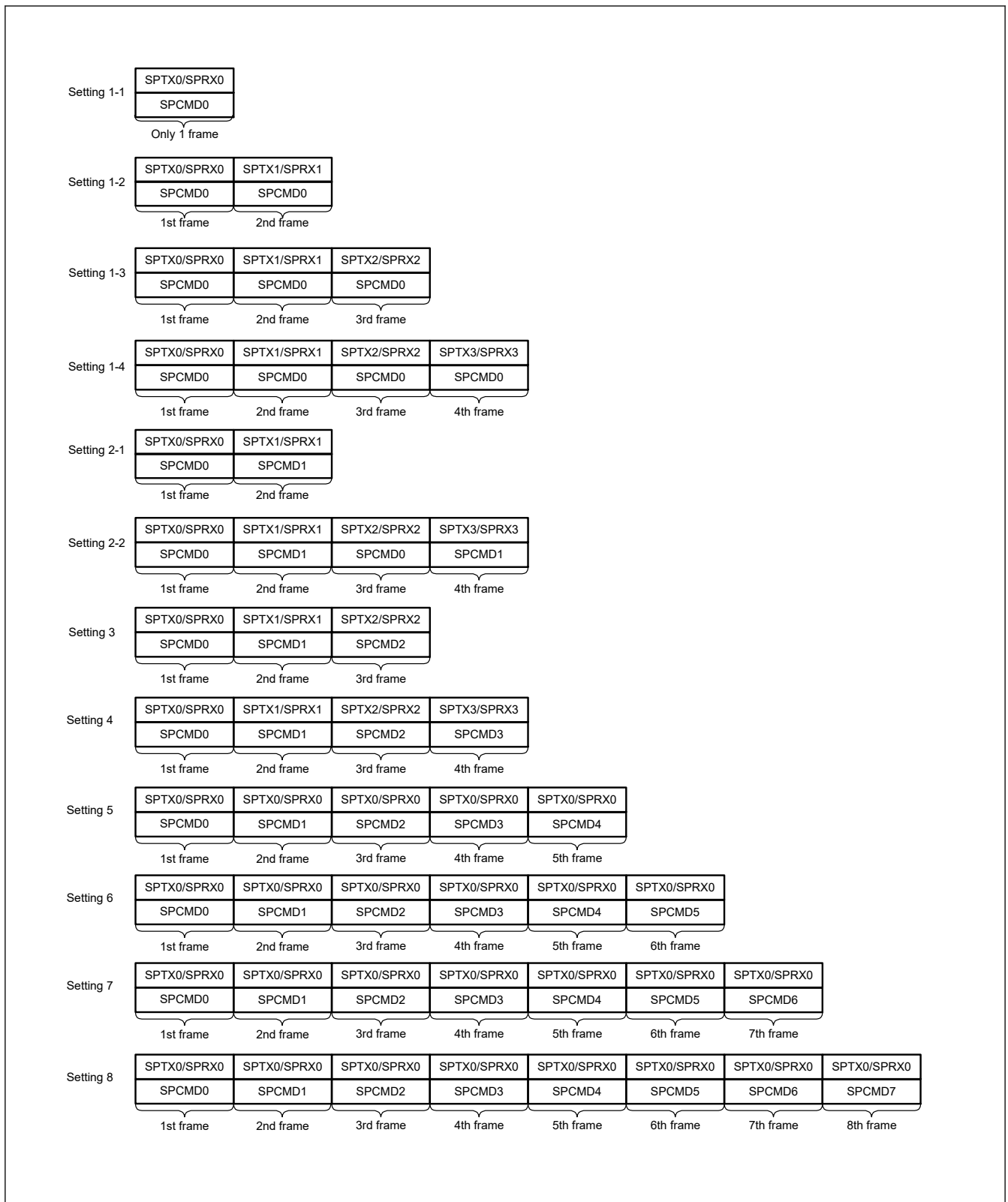
**Figure 34.54 Procedure for determining the form of a serial transfer in master mode**

In this section, a frame is the combination of the data in SPDR and the settings in SPCMDm.



**Figure 34.55 Conceptual diagram of frames**

Figure 34.56 shows the correspondence between the commands and the transmit and receive buffers in the sequence of operations specified by the settings.



**Figure 34.56 Correspondence between SPI Command Register and transmit and receive buffers in sequence operations**

**(4) Burst transfers**

This section describes burst transfer during transmit-receive/transmit-only operation.

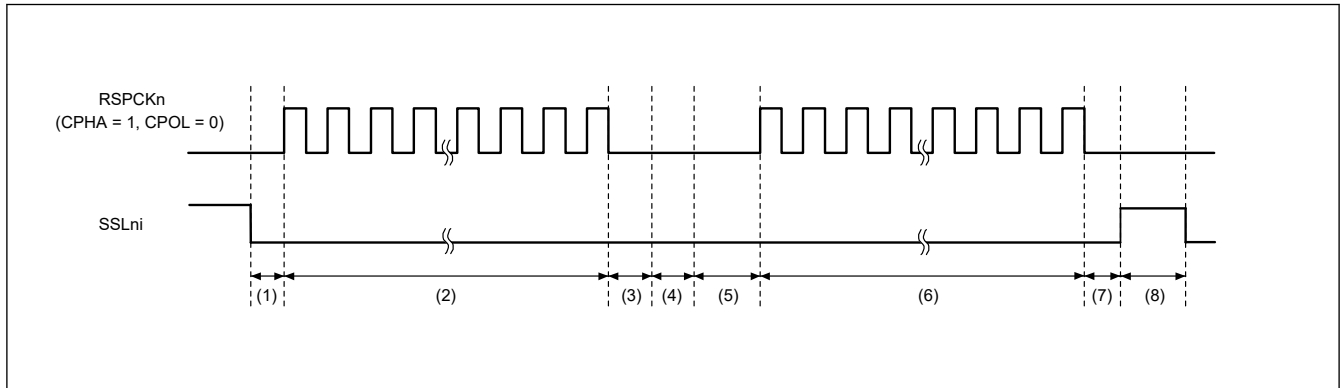
[In the Motorola-SPI case]

If the SPCMDm.SSLKP bit that the SPI references during the current serial transfer is 1, the SPI maintains the SSLni signal level during the serial transfer until the beginning of the SSLni signal assertion for the next serial transfer. If the SSLni signal level for the next serial transfer is the same as the SSLni signal level for the current serial transfer, the SPI can execute continuous serial transfers while keeping the SSLni signal assertion status (burst transfer).

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 0.

Figure 34.57 shows an example of an SSLni signal operation for a burst transfer that is implemented using the SPCMD0 and SPCMD1 register settings. This section describes SPI operations (1) to (8) shown in Figure 34.57.

Note: The polarity of the SSLni output signal depends on the SPCR3.SSLnP (n = 0 to 3) bits settings.



**Figure 34.57 Example of burst transfer operation using the SSLKP bit (BFDS = 0, SPFRF = 0)**

The SPI operation at times (1) to (8) in the figure is as follows:

1. Based on the SPCMD0 settings, the SPI asserts the SSLni signal and inserts RSPCK delays.
2. The SPI executes serial transfers in accordance with the SPCMD0 settings.
3. The SPI inserts an SSL negation delay.
4. Because the SPCMD0.SSLKP bit is 1, the SPI keeps the SSLni signal value specified in SPCMD0. This period additionally continues for 5 TCLK cycles (at minimum) that is the same as the next-access delay time of SPCMD0. If the shift register is empty after the passage of the minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
5. Based on the SPCMD1 settings, the SPI asserts the SSLni signal and inserts RSPCK delays.
6. The SPI executes serial transfers in accordance with the SPCMD1 settings.
7. Insert SSL negate delay.
8. Because the SPCMD1.SSLKP bit is 0, the SPI negates the SSLni signal. In addition, a next-access delay is inserted in accordance with SPCMD1.

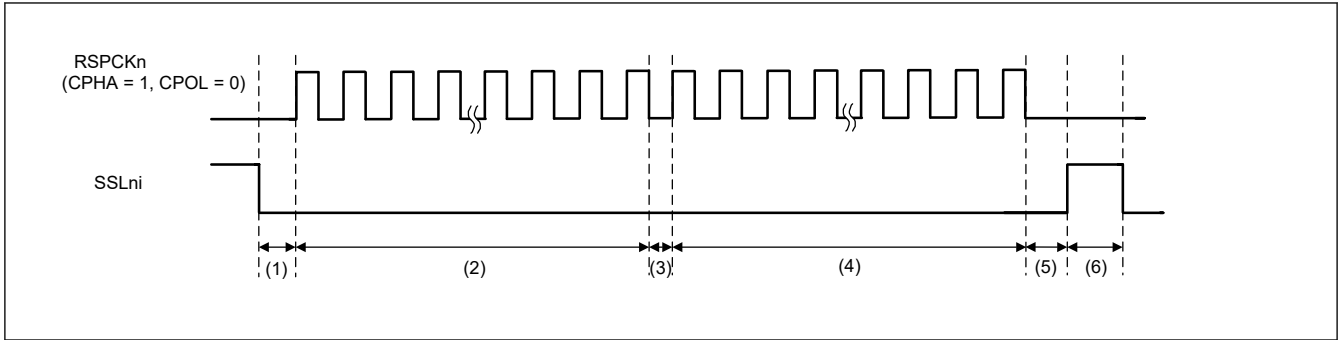
If the SSLni signal output settings in the SPCMDm register where 1 is assigned to the SSLKP bit are different from the SSLni signal output settings in the SPCMDm register to be used in the next transfer, the SPI switches the SSLni signal status to SSLni signal assertion as shown in (5) in Figure 34.57. This corresponds to the command for the next transfer.

Note: If such an SSLni signal switching occurs, the slaves that drive the MISO<sub>n</sub> signal compete, and collision of signal levels might occur.

The SPI in master mode references the SSLni signal operation within the module when the SSLKP bit is not used. When the SPCMDm.CPHA bit is 0, the SPI can accurately start serial transfers by using the SSLni signal assertion for the next transfer that is detected internally.

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 1.

Figure 34.58 shows an example of SSLni signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (6) shown in Figure 34.58. The SSLni output signal polarity depends on the set SPCR3.SSLnP (n = 0 to 3) value.



**Figure 34.58 Example of Burst Transfer Operation Using SSLKP Bit (BFDS = 1, SPFRF = 0)**

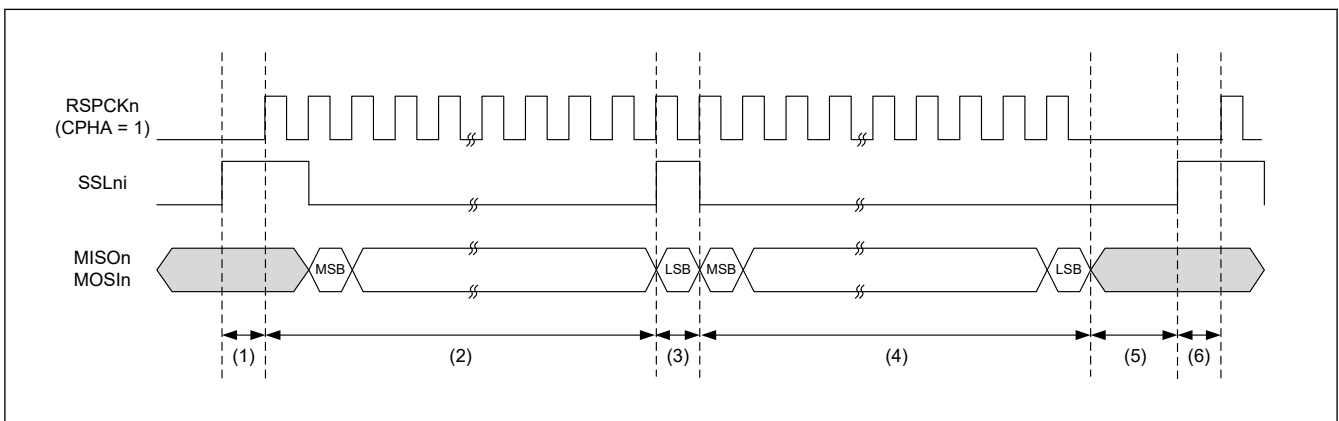
1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst transmission.
2. Perform serial transfer according to SPCMD0. Wait last clock until the next transmit data is stored in the shift register, if the shift register is empty during RSPCK negate period between frames.
3. The value of SSLni signal according to SPCMD0 was hold, because the SPCMD0.SSLKP bit is 1. RSPCK negate period between frames is 0.5RSPCK, if the shift register is not empty.
4. Perform serial transfer according to SPCMD1.
5. Insert SSLni negate delay for the last frame.
6. The SSLni signal is negated because the SSLKP bit in SPCMD1 is 0. Furthermore, the next-access delay is inserted according to SPCMD1.

[In the TI-SSP case]

SPI asserts the SSLni signal for one cycle at the start of serial transfer.

Serial transfer can be executed continuously by asserting the SSLni signal for one cycle at the start of the next serial transfer (burst transfer).

- When the SSLni signal level holding bit (SSLKP) of the SPI command register (SPCMD) is 1 and the burst transfer frame delay selection bit (BFDS) of the SPI control register (SPCR) is 1, SPCMD0 to SPCMD1 are shown in [Figure 34.59](#). The following shows an example of SSLni signal operation and serial data MISO<sub>n</sub>/MOSI<sub>n</sub> when burst transfer is realized using the settings. The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.



**Figure 34.59 Example of Burst Transfer Operation (SPFRF = 1)**

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst transmission.
2. Perform serial transfer according to SPCMD0.
3. Final data transfer and SSLni assertion are performed simultaneously. If the shift register is empty during the RSPCK negation period between frames, wait for the output of the last clock until the transmission data for the next transfer is stored in the shift register.

4. Perform serial transfer according to SPCMD1.
5. Insert OE negate delay for the last frame.
6. Insert the next access delay according to SPCMD1.

If the SSLni signal output setting in SPCMD with the SSLKP bit set to 1 differs from the SSLni signal output setting in SPCMD to be used for the next transfer, the SPI changes the SSLni signal state when the SSLni signal corresponding to the next-transfer command is asserted ((5)). Note that, if an SSLni signal change like this takes place, slaves that drive the MISO<sub>n</sub> signal may conflict with each other, which may cause collision of signal level.

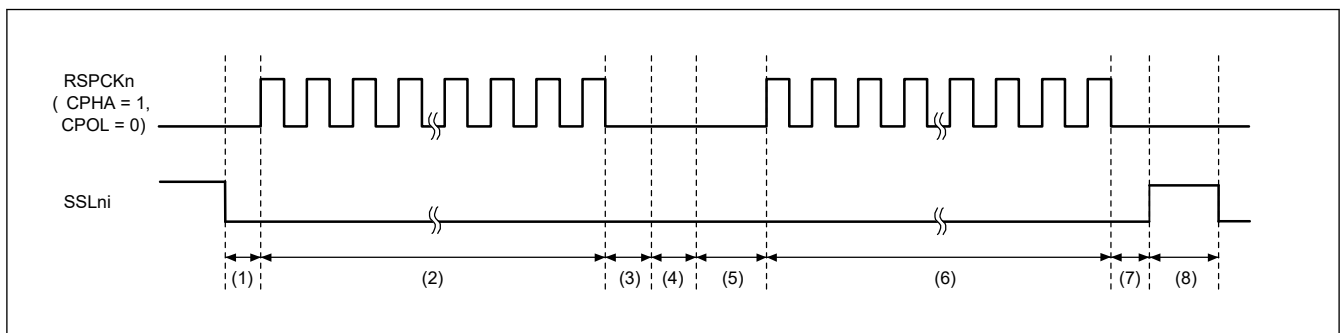
This section describes burst transfer during receive-only operation.

[In the Motorola-SPI case]

When the SSLKP bit in the SPI command register (SPCMD), which the SPI references in the current serial transfer, is 1, the SPI retains the SSLni signal level during serial transfer until the SSLni signal assertion of the next serial transfer starts. When the SSLni signal level in the next serial transfer is the same as the SSLni signal level in the current serial transfer, the SPI can continuously perform serial transfer while holding the SSLni signal assertion status (burst transfer).

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 0.

Figure 34.60 shows an example of SSLni signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (8) shown in Figure 34.60. The SSLni output signal polarity depends on the set SPI SSLni signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.



**Figure 34.60 Example of Burst Transfer Operation Using SSLKP Bit (BFDS = 0, SPFRF = 0)**

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0.
2. Perform serial transfer according to SPCMD0.
3. Insert an SSLni negation delay.
4. The SSLni signal value in SPCMD0 is retained because the SSLKP bit in SPCMD0 is 1. This period additionally continues for 5 TCLK cycles (at minimum) that is the same as the next-access delay time of SPCMD0.
5. Assert the SSLni signal and insert an RSPCK delay according to SPCMD1.
6. Perform serial transfer according to SPCMD1.
7. Insert SSLni negate delay.
8. The SSLni signal is negated because the SSLKP bit in SPCMD1 is 0. Furthermore, the next-access delay is inserted according to SPCMD1.

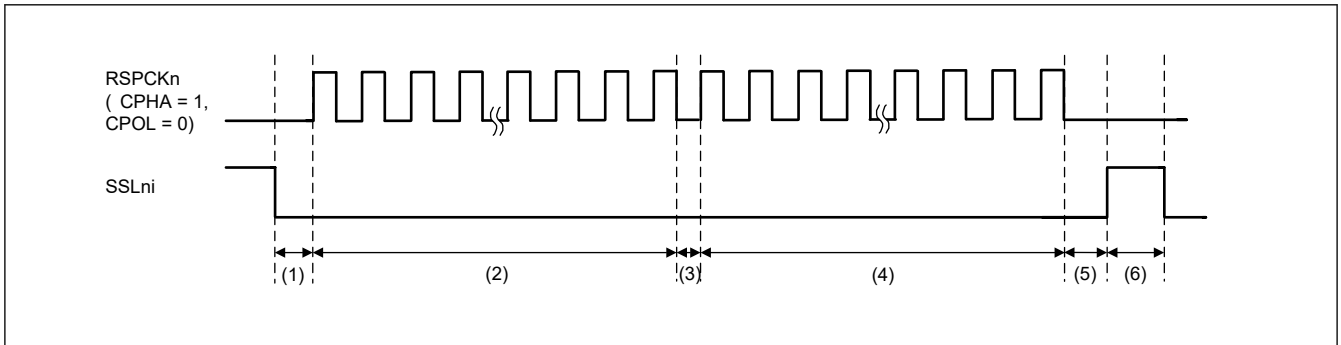
If the SSLni signal output setting and the SSLni signal output setting between SPCMDs used for burst transfer are different, SPI switches the SSLni signal state when the SSLni signal corresponding to the next transfer command is asserted (5). Note that, if an SSLni signal change like this takes place, slaves that drive the MISO<sub>n</sub> signal may conflict with each other, which may cause collision of signal level.

The SPI in master mode references the SSLni signal operation in the module when SSLKP is not used.

Even when the CPHA bit in SPCMD is 0, the SPI can accurately start serial transfer by using the next transfer SSLni signal assertion detected internally. For this reason, burst transfer in master mode is enabled regardless of the set CPHA bit value. (See section 34.3.11. Initializing the SPI.)

- When Between Burst Transfer Frames Delay Select bit (BFDS) of SPI control register (SPCR) is 1.

Figure 34.61 shows an example of SSLni signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (6) shown in Figure 34.61. The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.



**Figure 34.61 Example of Burst Transfer Operation Using SSLKP Bit (BFDS = 1, SPFRF = 0)**

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst transmission.
2. Perform serial transfer according to SPCMD0.
3. Since it is not the last frame, the SSLni signal value at SPCMD0 is retained. RSPCKn negation between frames is 0.5 RSPCKn for the next frame.
4. Perform serial transfer according to SPCMD1.
5. Insert SSLni negate delay for the last frame.
6. The SSLni signal is negated. Furthermore, the next-access delay is inserted according to SPCMD1.

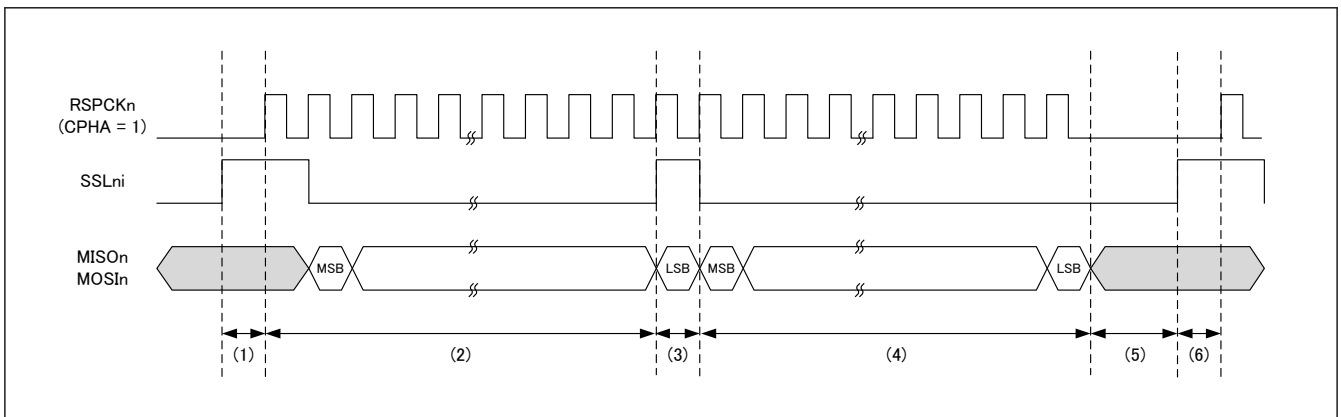
Note: Last frame: Frame set by RMFM[4:0] bits when SPCR2.RMFM[4:0] ≠ 0x00  
 Or, a frame in which SPCR2.RMEDTG = 1 has been accepted.

[In the TI-SSP case]

SPI asserts the SSLni signal for one cycle at the start of serial transfer.

Serial transfer can be executed continuously by asserting the SSLni signal for one cycle at the start of the next serial transfer (burst transfer).

- When the SSLni signal level holding bit (SSLKP) of the SPI command register (SPCMD) is 1 and the burst transfer frame delay selection bit (BFDS) of the SPI control register (SPCR) is 1, SPCMD0 to SPCMD1 are shown in Figure 34.62. The following shows an example of SSLni signal operation and serial data MISO<sub>n</sub> / MOSI<sub>n</sub> when burst transfer is realized using the settings. The SSLni output signal polarity depends on the set SPI SSLi signal polarity bit (SPCR3.SSLiP) (i = 0 to 3) value.



**Figure 34.62 Example of Burst Transfer Operation (SPFRF = 1)**

1. Assert the SSLni signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst transmission.



2. Perform serial transfer according to SPCMD0.
3. Final data transfer and SSLni assertion are performed simultaneously.
4. Perform serial transfer according to SPCMD1.
5. Insert OE negate delay for the last frame.
6. Insert the next access delay according to SPCMD1.

Note: Last frame: Frame set by RMFM[4:0] bits when SPCR2.RMFM[4:0]  $\neq$  0x00  
Or, a frame in which SPCR2.RMEDTG = 1 has been accepted.

If the SSLni signal output setting between the SPCMDs used for burst transfer differs from the SSLni signal output setting, SPI switches the SSLni signal state when the SSLni signal corresponding to the next transfer command is asserted (5). Note that, if an SSLni signal change like this takes place, slaves that drive the MISO<sub>n</sub> signal may conflict with each other, which may cause collision of signal level.

#### (5) RSPCK delay (t1)

The RSPCK delay value of the SPI in master mode depends on the SPCMDm.SCKDEN bit setting and the SPDECR.SCKDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced during a serial transfer by pointer control, and determines an RSPCK delay using the SPCMDm.SCKDEN bit and SPDECR.SCKDL[2:0] bits, as listed in [Table 34.10](#). For a definition of RSPCK delay, see [section 34.3.5. Transfer Formats](#).

RSPCK delay insert to only the first frame of burst transmission, when transmit without “Between Burst Transfer Frames Delay”. (The SPCMD.SSLKP bit is 1 and the SPCR.BFDS bit is 1.)

**Table 34.10 Relationship between the SPCMDm.SCKDEN bit, SPDECR.SCKDL[2:0] bits, and RSPCK delay**

SPCMDm.SCKDEN bit	SPDECR.SCKDL[2:0] bits	RSPCK delay	
		Motorola-SPI	TI-SSP
0	000b to 111b	1 RSPCK	0 RSPCK
1	000b	1 RSPCK	1 RSPCK
	001b	2 RSPCK	2 RSPCK
	010b	3 RSPCK	3 RSPCK
	011b	4 RSPCK	4 RSPCK
	100b	5 RSPCK	5 RSPCK
	101b	6 RSPCK	6 RSPCK
	110b	7 RSPCK	7 RSPCK
	111b	8 RSPCK	8 RSPCK

#### (6) SSL negation delay (t2)

The SSL negation delay value of the SPI in master mode depends on the SPCMDm.SLN DEN bit setting and the SPDECR.SLN DL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced by pointer control during a serial transfer, and determines an SSL negation delay using the SPCMDm.SLN DEN bit and SPDECR.SLN DL[2:0] bits, as listed in [Table 34.11](#). For a definition of SSL negation delay, see [section 34.3.5. Transfer Formats](#).

An SSL negation delay is inserted to only the last frame of the burst transmission, that is, transmit without between burst transfer frames delay. (SPCMD.SSLKP bit is 1 and SPCR.BFDS bit is 1).

**Table 34.11 Relationship between the SPCMDm.SLN DEN bit, SPDECR.SLN DL[2:0] bits, and SSL negation delay (1 of 2)**

SPCMDm.SLN DEN bit	SPDECR.SLN DL[2:0] bits	SSL negation delay
0	000b to 111b	1 RSPCK

**Table 34.11 Relationship between the SPCMDm.SLN DEN bit, SPDECR.SLN DL[2:0] bits, and SSL negation delay (2 of 2)**

SPCMDm.SLN DEN bit	SPDECR.SLN DL[2:0] bits	SSL negation delay
1	000b	1 RSPCK
	001b	2 RSPCK
	010b	3 RSPCK
	011b	4 RSPCK
	100b	5 RSPCK
	101b	6 RSPCK
	110b	7 RSPCK
	111b	8 RSPCK

**(7) Next-access delay (t3)**

The next-access delay value of the SPI in master mode depends on the SPCMDm.SPNDEN bit setting and the SPDECR.SPNDL[2:0] bits setting. The SPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and then determines a next-access delay during serial transfer using the SPCMDm.SPNDEN bit and SPDECR.SPNDL[2:0] bits, as listed in [Table 34.12](#). For a definition of next-access delay, see [section 34.3.5. Transfer Formats](#).

A next-Access delay is inserted to only the last frame of the burst transmission, that is, transmit without between burst transfer frames delay. (SPCMD.SSLKP bit is 1 and SPCR.BFDS bit is 1).

**Table 34.12 Relationship between the SPCMDm.SPNDEN bit, SPDECR.SPNDL[2:0] bits, and next-access delay**

SPCMDm.SPNDEN bit	SPDECR.SPNDL[2:0] bits	Next-access delay
0	000b to 111b	1 RSPCK + 5 TCLK
1	000b	1 RSPCK + 5 TCLK
	001b	2 RSPCK + 5 TCLK
	010b	3 RSPCK + 5 TCLK
	011b	4 RSPCK + 5 TCLK
	100b	5 RSPCK + 5 TCLK
	101b	6 RSPCK + 5 TCLK
	110b	7 RSPCK + 5 TCLK
	111b	8 RSPCK + 5 TCLK

**(8) Initialization flow**

[Figure 34.63](#) shows an example of SPI initialization flow when the SPI is in master mode. For information on how to set up the Interrupt Controller Unit (ICU), DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.

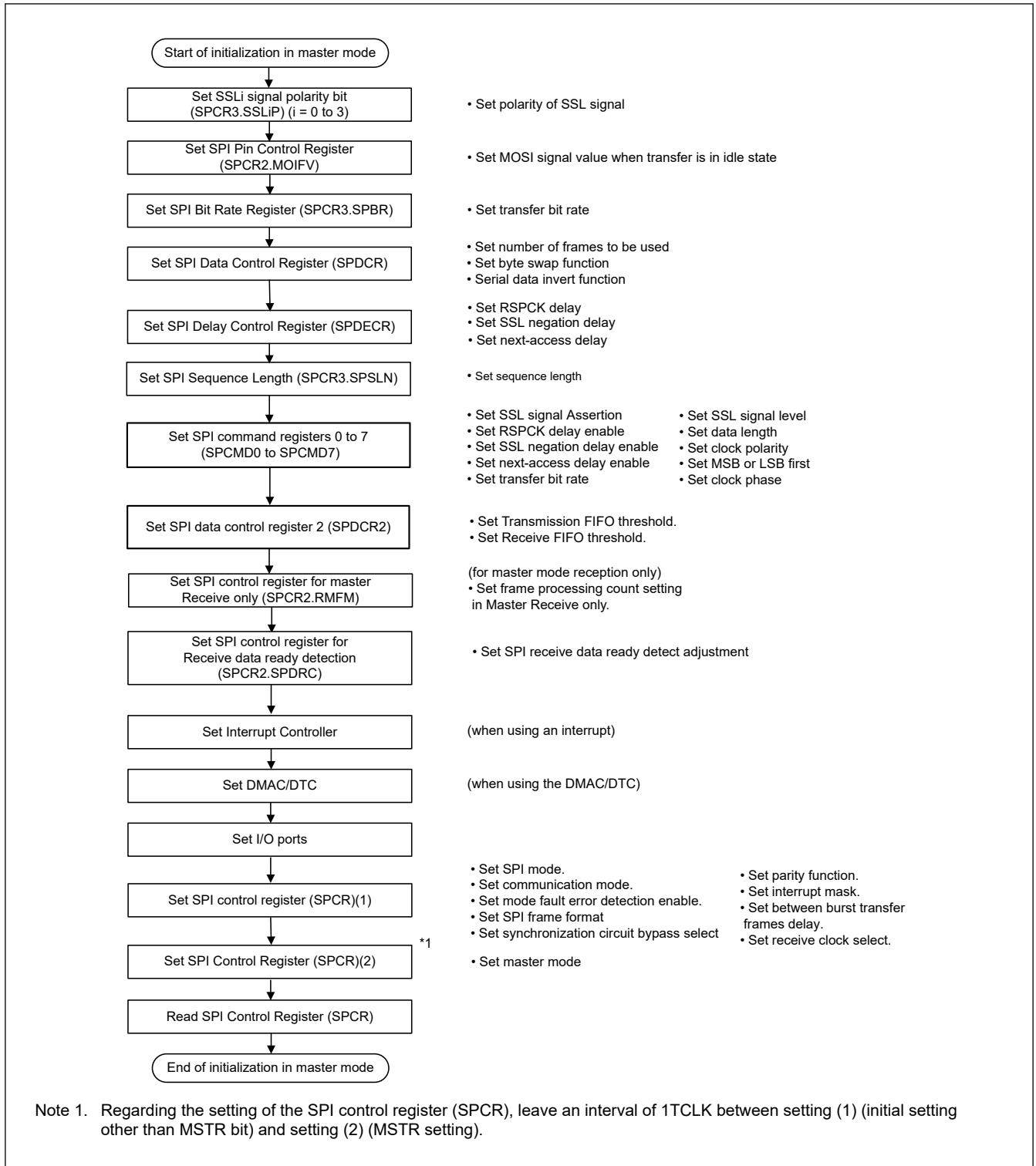


Figure 34.63 Example of initialization flow in master mode for SPI operation

(9) Software processing flow

Figure 34.64 to Figure 34.67 show examples of the software processing flow.

**Transmit processing flow**

When transmitting data, with the SPIi\_SPII or SPIi\_SPCEND interrupt enabled, the CPU is notified of the completion of data transmission after the last data write for transmission.

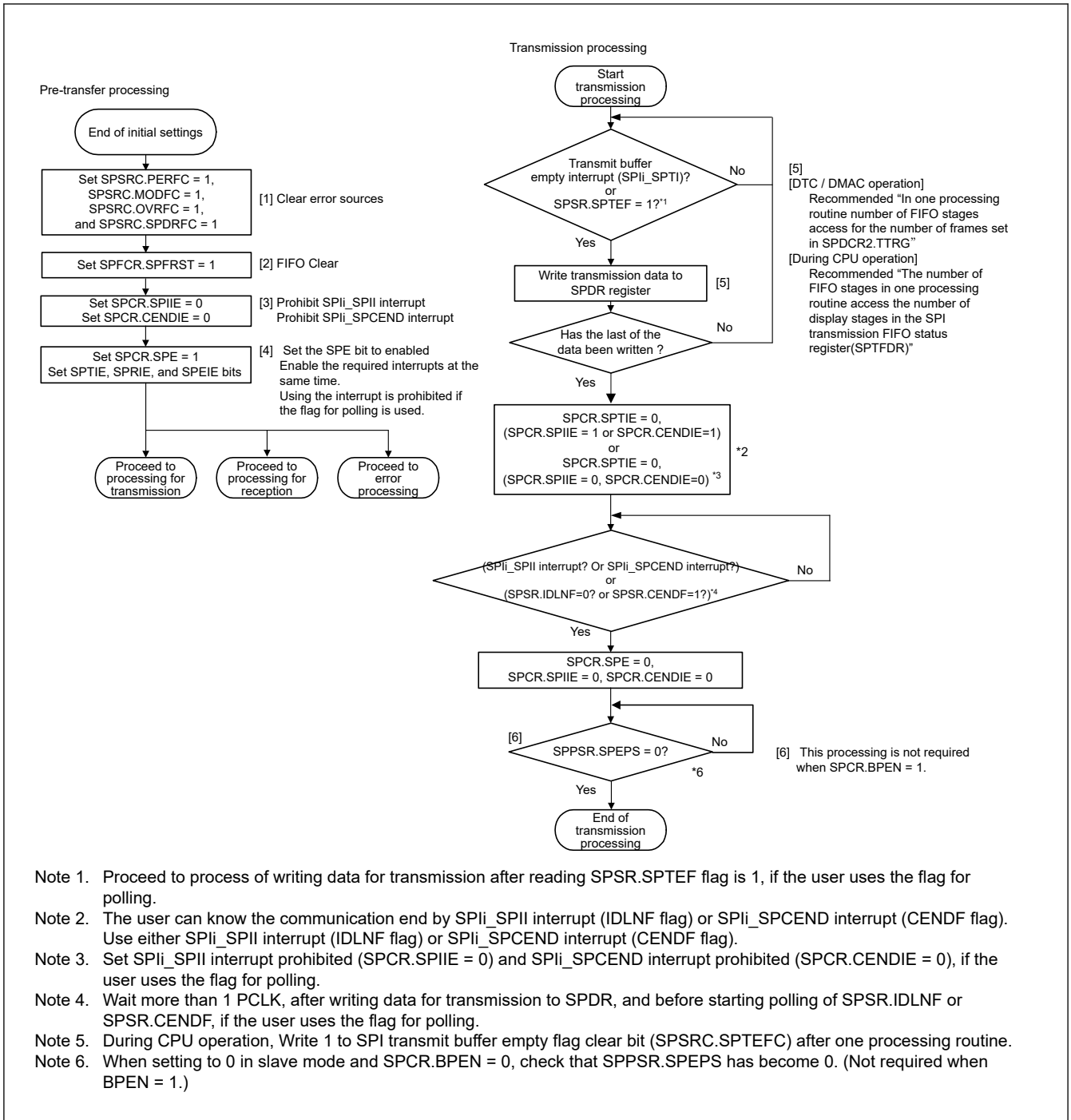


Figure 34.64 Transmission flow in master mode

**Receive processing flow**

The SPI has receive only operation in slave mode.

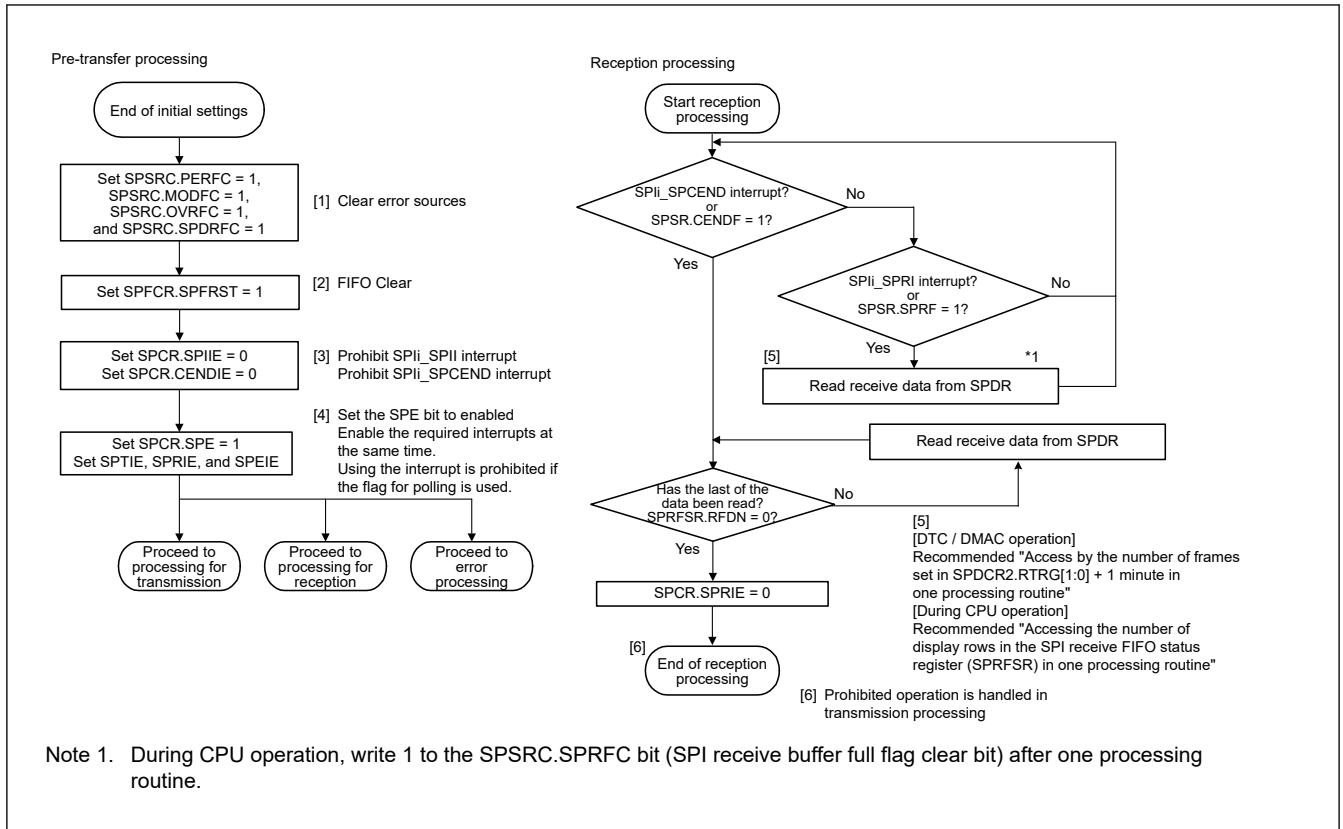


Figure 34.65 Reception flow in master mode

Receive only processing flow in master mode

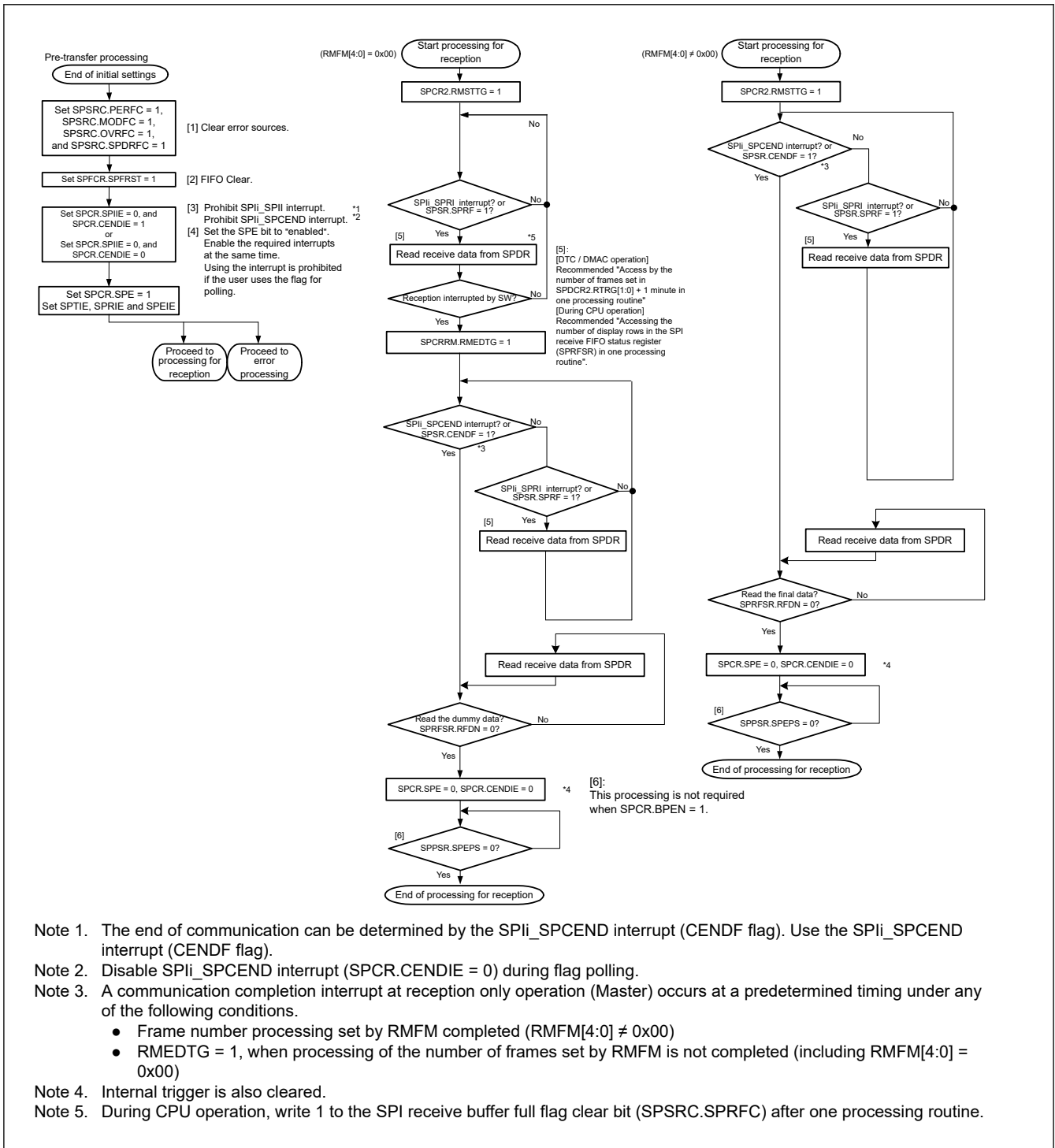


Figure 34.66 Software Processing Flowchart in Master Mode (Reception-only)

Error processing flow

The SPI detects the following errors:

- Mode fault error
- Underrun error
- Overrun error
- Parity error

When a mode fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, the SPCR.SPE bit is not cleared and operations for transmission and reception continue. Therefore, Renesas recommends clearing the SPCR.SPE bit to stop operations for errors other than mode fault errors. Not doing so leads to updating of the SPDCR2.SPECM[2:0] bits.

When an error is detected using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPI<sub>i</sub>\_SPTI or SPI<sub>i</sub>\_SPRI interrupt request. If the SPI<sub>i</sub>\_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

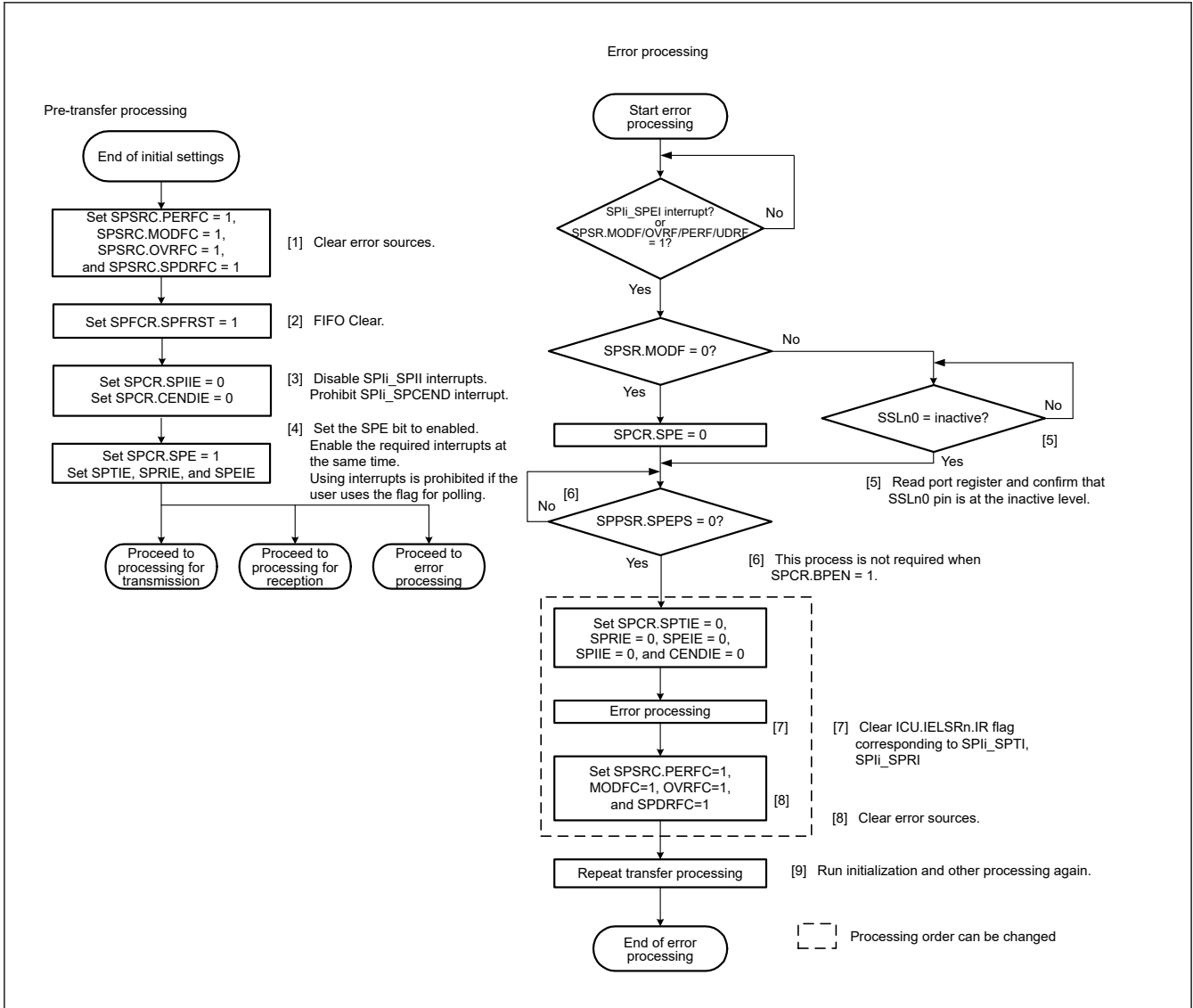


Figure 34.67 Error processing flow in master mode

### 34.3.12.2 Slave mode operation

#### (1) Starting a serial transfer

When the SPCMD0.CPHA bit is 0, if the SPI detects an SSL<sub>n</sub>0 input signal assertion, it must drive valid data to the MISO<sub>n</sub> output signal. For this reason, when the CPHA bit is 0, the assertion of the SSL<sub>n</sub>0 input signal triggers the start of a serial transfer.

When the CPHA bit is 1, if the SPI detects the first RSPCK<sub>n</sub> edge in an SSL<sub>n</sub>0 signal asserted condition, it must drive valid data to the MISO<sub>n</sub> output signal. For this reason, when the CPHA bit is 1, the first RSPCK<sub>n</sub> edge in an SSL<sub>n</sub>0 signal asserted condition triggers the start of a serial transfer.

Regardless of the CPHA bit setting, the SPI drives the MISO<sub>n</sub> output signal on SSL<sub>n</sub>0 signal assertion. The data that is output by the SPI is either valid or invalid, depending on the CPHA bit setting.

For details on the SPI transfer format, see [section 34.3.5. Transfer Formats](#). The polarity of the SSLn0 input signal depends on the SPCR3.SSL0P setting.

## (2) Terminating a serial transfer

Regardless of the SPCMD0.CPHA bit setting, the SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO is less than the number of FIFO stages, on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR register. On termination of a serial transfer, the SPI changes the status of the shift register to empty, regardless of the receive buffer state. A mode fault error occurs if the SPI detects an SSLn0 input signal negation from the beginning of serial transfer to the end of serial transfer (see [section 34.3.10. Error Detection](#)).

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length is determined by the SPCMD0.SPB[4:0] bits setting. The polarity of the SSLn0 input signal is determined by the SPCR3.SSL0P bit setting. For details on the SPI transfer format, see [section 34.3.5. Transfer Formats](#).

## (3) Notes on single-slave operations

[In the Motorola-SPI case]

If the SPCMD0.CPHA bit is 0, the SPI starts serial transfers when it detects the assertion edge for an SSLn0 input signal. In the configuration shown in [Figure 34.8](#), if the SPI is used in single-slave mode, the SSLn0 signal is fixed at an active state. Therefore, when the CPHA bit is set to 0, the SPI cannot correctly start a serial transfer. For the SPI to correctly execute transmit and receive operations in slave mode when the SSLn0 input signal is fixed at an active state, the CPHA bit must be set to 1. Do not fix the SSLn0 input signal if there is a requirement for setting the CPHA bit to 0.

[In the TI-SSP case]

When SPI is used as a single slave in the configuration shown in the [Figure 34.8](#), the SSLn0 input signal is always fixed to the inactive state, so the SPI cannot start the serial transfer.

When using a single slave, use the configuration shown in the example in [Figure 34.7](#).

## (4) Burst transfer

[In the Motorola-SPI case]

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLn0 input signal. When the CPHA bit is 1, the serial transfer period is the period from the first RSPCKn edge to the sampling timing for the reception of the final bit in an SSLn0 signal active state. Even when the SSLn0 input signal remains at the active level, the SPI can accommodate burst transfers, because it can detect the start of an access.

When the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

[In the TI-SSP case]

In serial transfer, data transfer starts after the SSLn0 input signal is asserted for RSPCK 1 cycle. Since frame transfer starts from the SSLn0 input signal, SSLn0 must be asserted between frames.

## (5) Initialization flow

[Figure 34.68](#) shows an example of initialization flow for SPI operation when the SPI is in slave mode. For a description of how to set up the ICU, DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.



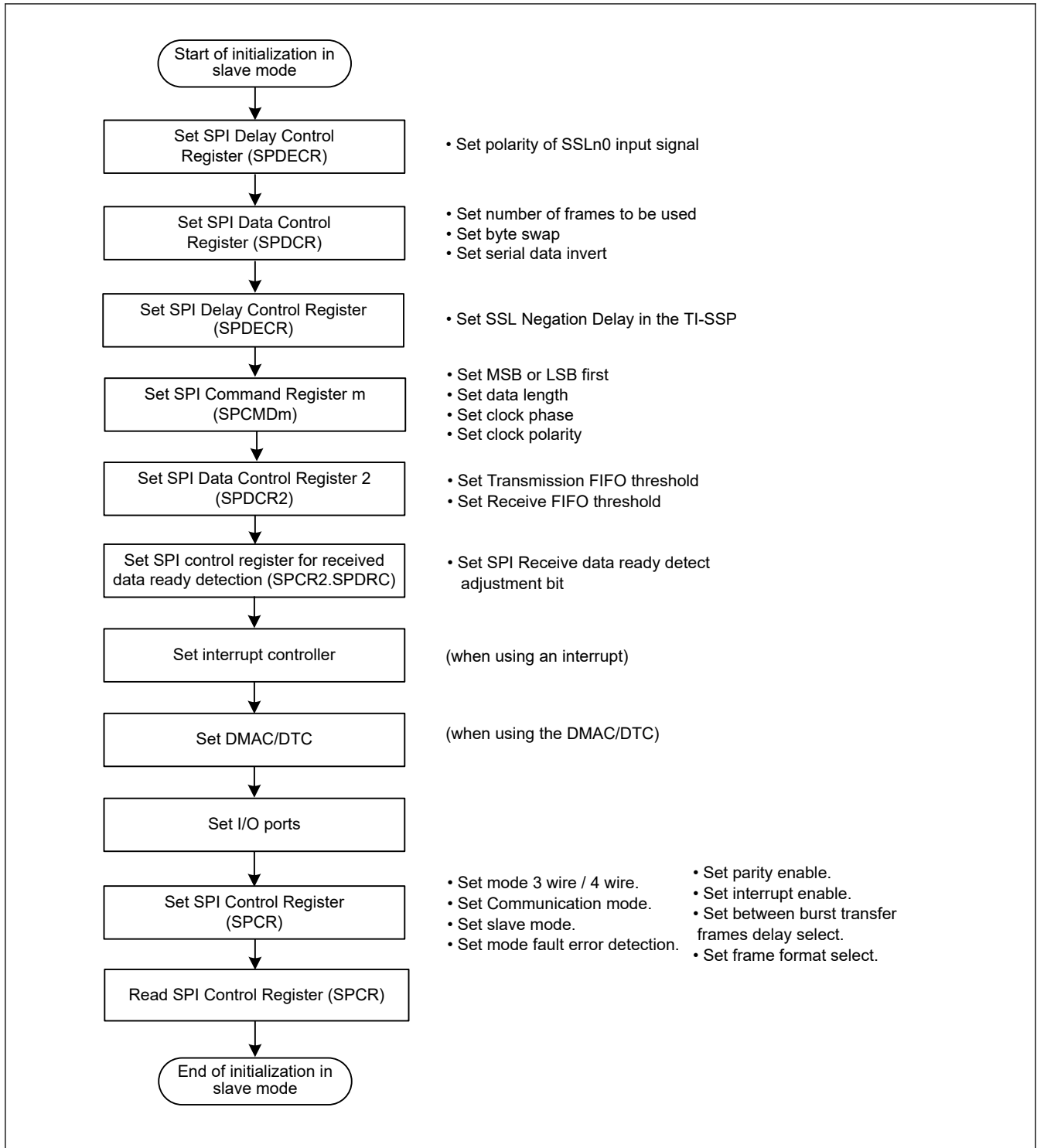


Figure 34.68 Example initialization flow in slave mode for SPI operation

(6) Software processing flow

Figure 34.69 to Figure 34.72 show examples of the flow of software processing.

Transmit processing flow

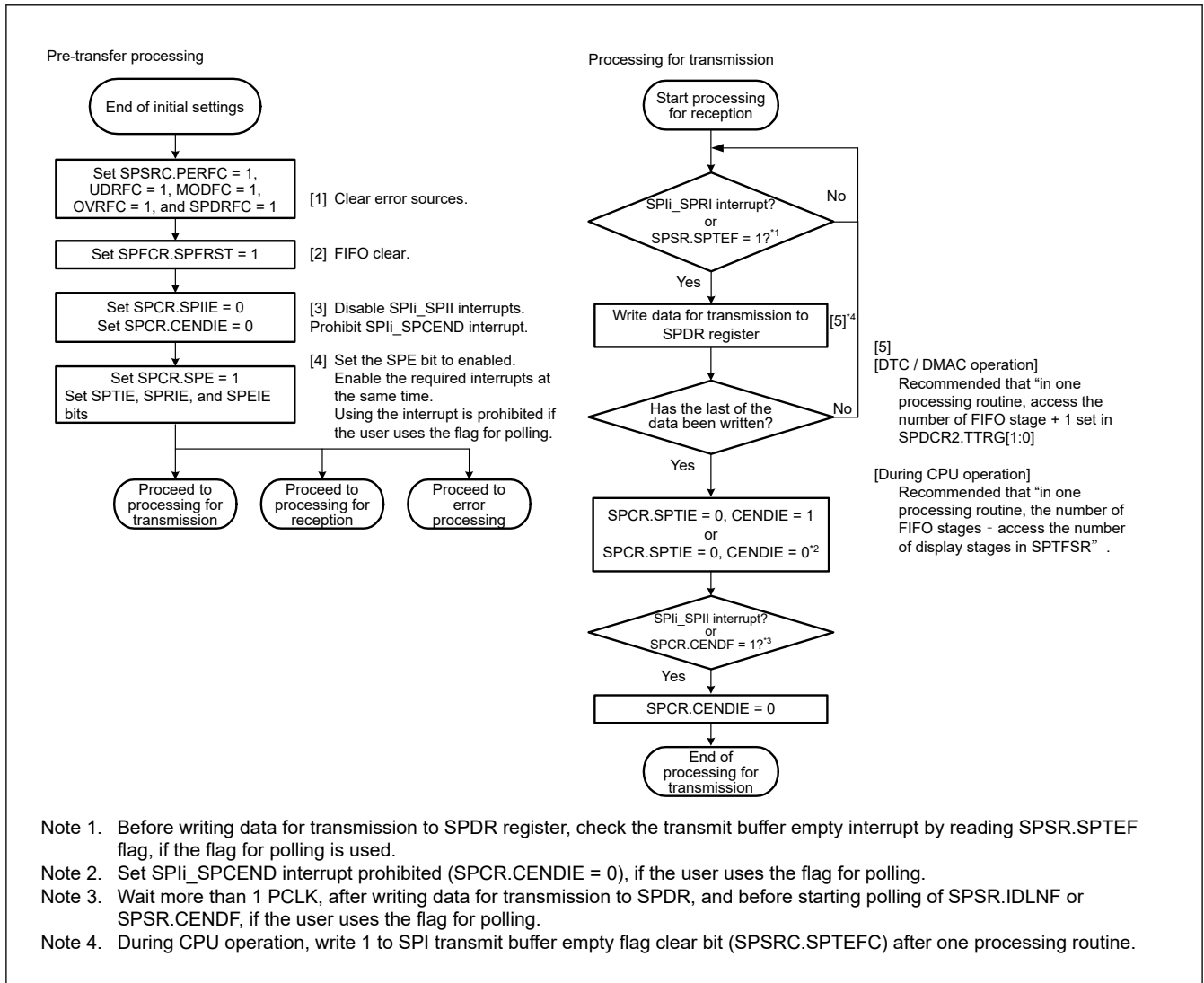


Figure 34.69 Transmission flow in slave mode

Receive processing flow

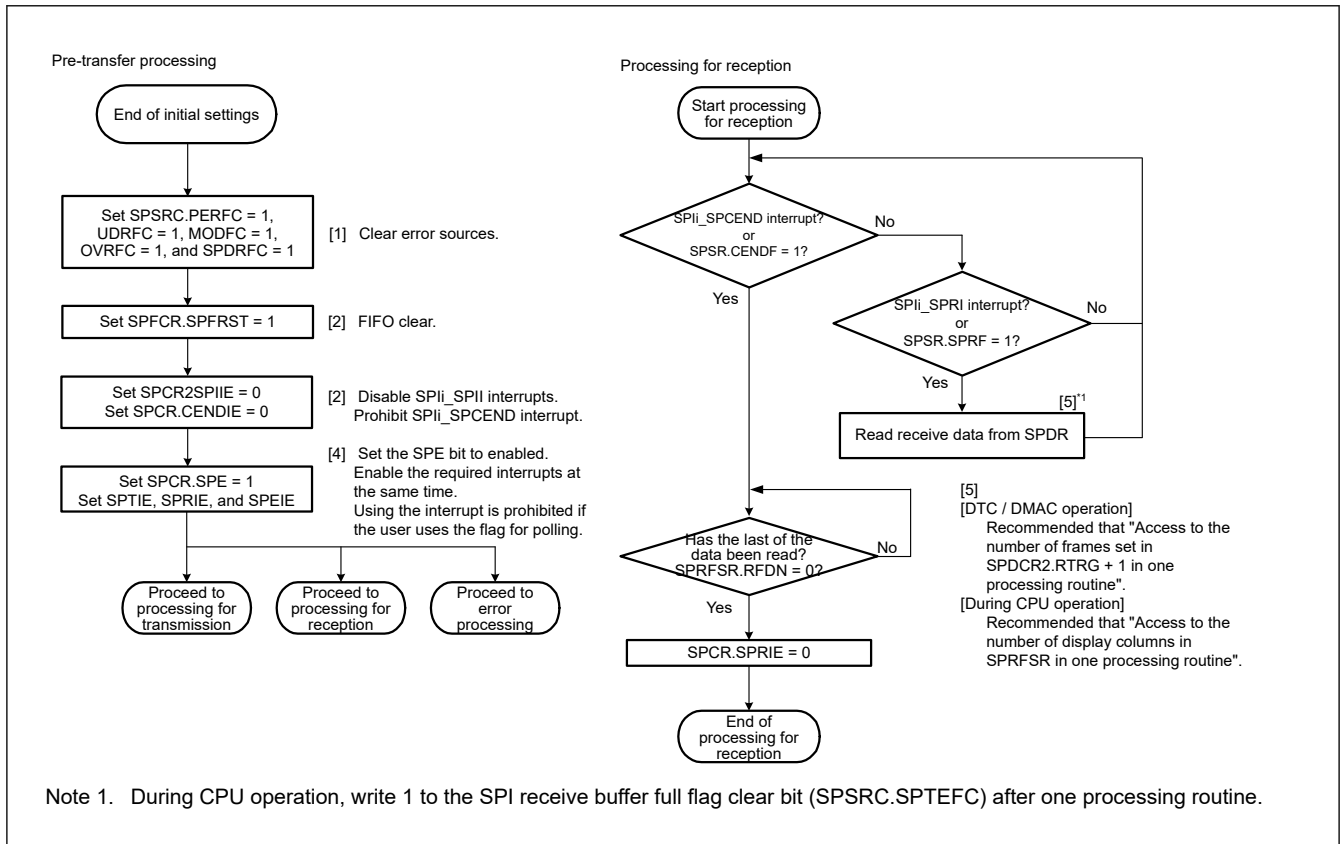
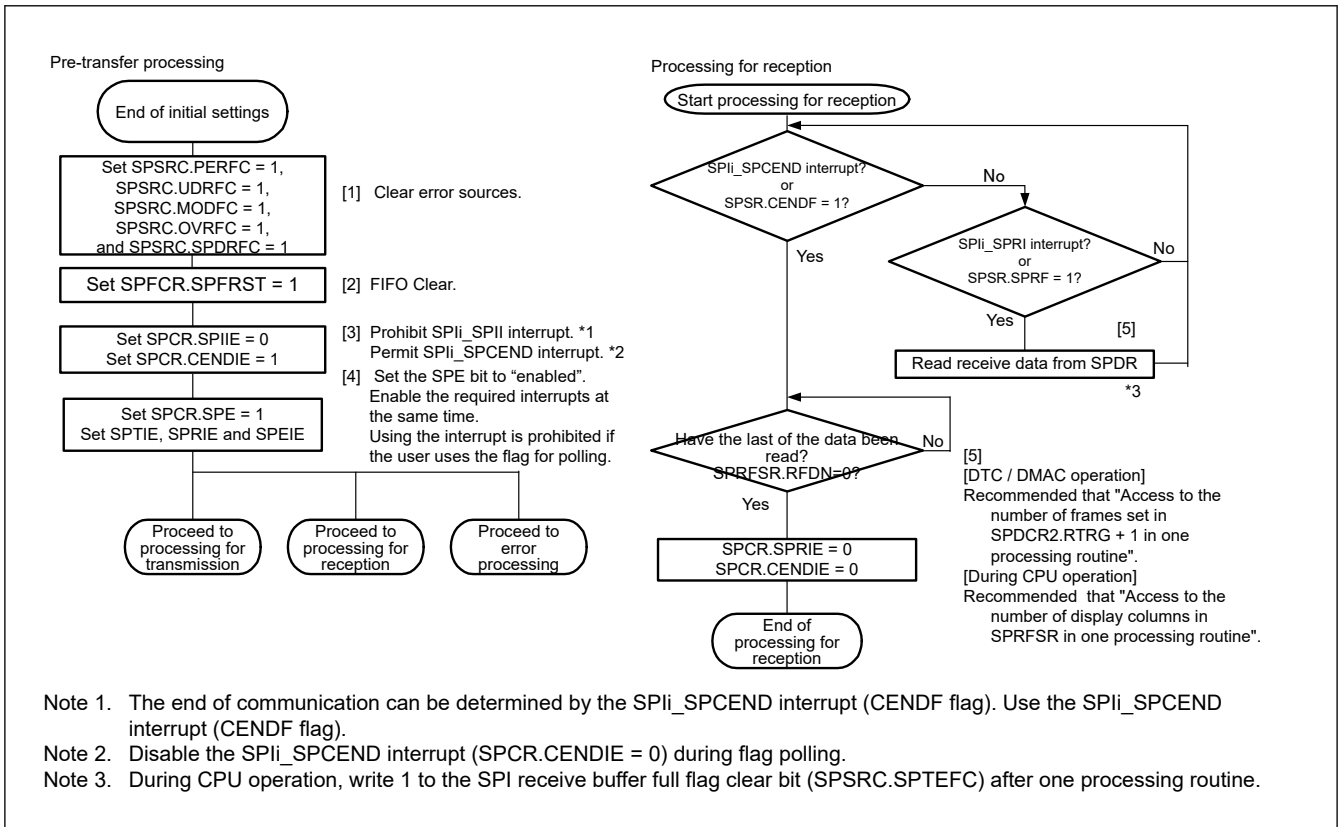


Figure 34.70 Reception flow in slave mode

**Master Reception-only processing flow**



**Figure 34.71 Software Processing Flowchart in Master Mode (Reception-only)**

**Error processing flow**

In slave mode operation, even when a mode fault error is generated, the SPSR.MODF flag can be cleared regardless of the state of the SSLn0 pin.

When an error is detected by using an interrupt, clear the ICU.IELSRn.IR flag in the error processing routine. If this is not done, the ICU.IELSRn.IR flag might continue to indicate the SPIi\_SPTI or SPIi\_SPRI interrupt request. If the SPIi\_SPRI interrupt request is indicated, read the receive buffer and initialize the sequencer in the SPI.

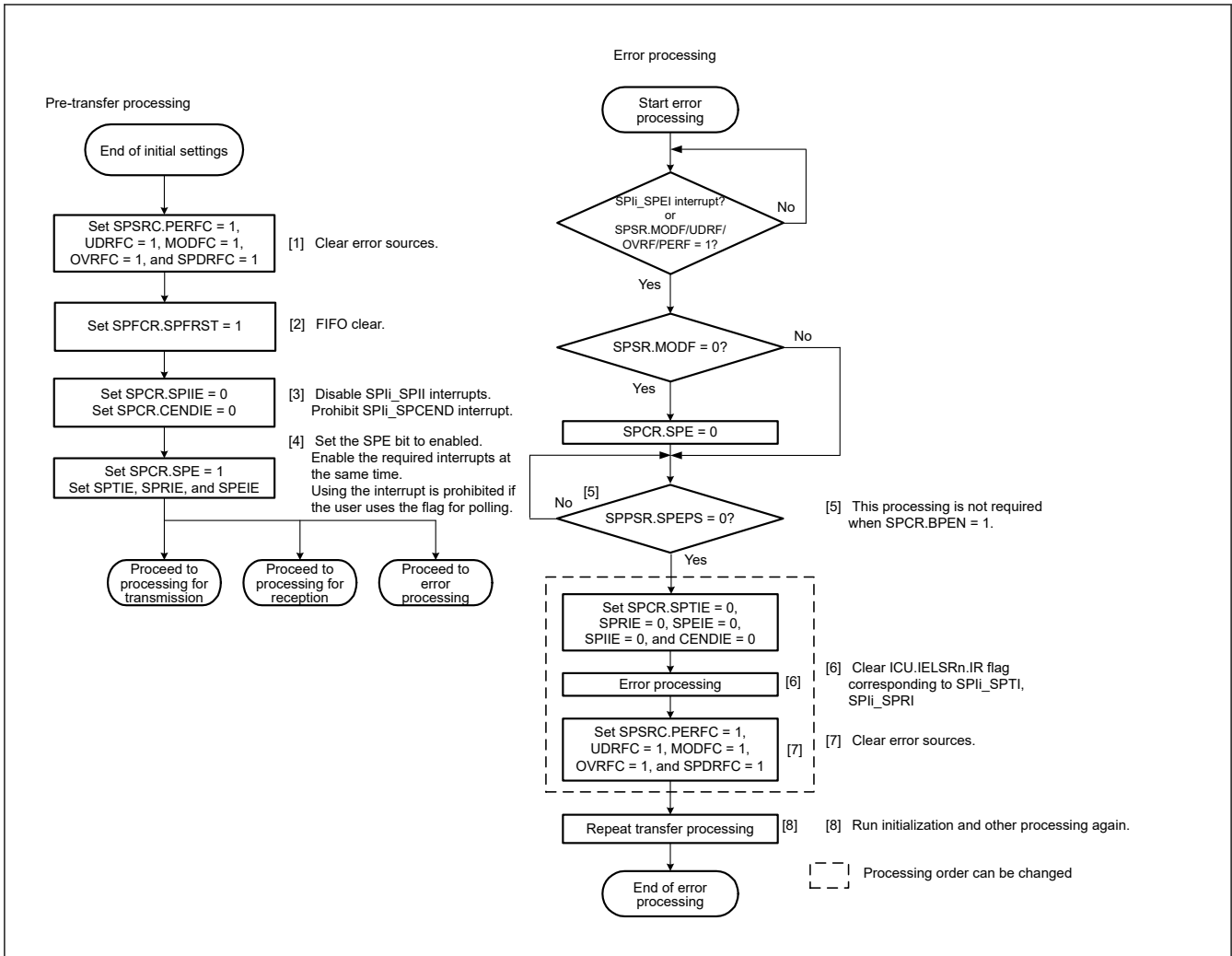


Figure 34.72 Error processing flow for slave mode

### 34.3.13 Clock Synchronous Operation

Setting the SPCR.SPMS bit to 1 selects clock synchronous operation of the SPI. In clock synchronous operation, the SSL<sub>n</sub>i pin is not used, and the RSPCK<sub>n</sub>, MOS<sub>n</sub>, and MISO<sub>n</sub> pins handle communications. All SSL<sub>n</sub>i pins are available as I/O port pins.

Although clock synchronous operation does not require the use of the SSL<sub>n</sub>i pin, operation of the module is the same as in SPI operation. In both master mode and slave mode operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected, because the SSL<sub>n</sub>i pin is not used.

Additionally, do not perform operation if clock synchronous operation is enabled when the SPCMD<sub>m</sub>.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

#### 34.3.13.1 Master mode operation

##### (1) Starting serial transfer

When data is written to the SPI data register (SPDR) while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTX<sub>n</sub>, n = 0 to 3) data in SPDR. While the shift register is empty, the SPI copies transmit buffer data to the shift register to start serial transfer. After the SPI copies transmit data to the shift register, it changes the shift register status to full. Upon completion of serial transfer, the SPI changes the shift register status to empty. The shift register status cannot be monitored.

For details about the SPI transfer format, see section 34.3.5. Transfer Formats. In clock synchronous operation, however, the SSL<sub>n</sub>0 output signal is not used for communication.

(2) Terminating serial transfer

The SPI terminates the serial transfer after transmitting an RSPCKn edge corresponding to the sampling timing. If the number of data stored in the receive FIFO < the number of FIFO stages, on termination of serial transfer, the SPI copies data from the shift register to the receive buffer of the SPI Data Register (SPDR).

The final sampling timing varies depending on the bit length of transfer data. In master mode, the SPI data length depends on the SPCMDm.SPB[4:0] bits setting. Transfer in clock synchronous operation is conducted without the SSLn0 output signal. For details on the SPI transfer format, see [section 34.3.5. Transfer Formats](#).

(3) Sequence control

The transfer format used in master mode is determined by the SPCR3, SPCMDm, and SPDECR registers. Although the SSLni signals are not output in clock synchronous operation, these settings are valid.

The SPCR3.SPSSLN[2:0] bits determine the sequence configuration for serial transfers that are executed by the SPI in master mode. The following parameters are specified in the SPCMDm register:

- SSLni output signal value
- MSB or LSB first
- Data length
- Some of the bit rate settings
- RSPCKn polarity and phase
- Whether SPDECR.SCKDL is to be referenced
- Whether SPDECR.SLNDL is to be referenced
- Whether SPDECR.SPNDL is to be referenced

SPCR3.SPBR holds some of the bit rate settings such as SPDECR.SCKDL, an SPI clock delay value, SPDECR.SLNDL, an SSL negation delay, and SPDECR.SPNDL, a next-access delay value.

Based on the sequence length that is assigned to SPCR3, the SPI makes up a sequence comprised of a part or all of SPCMDm register. The SPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPDCR2.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the SPI function is enabled, the SPI loads the pointer to the commands in SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer. The SPI increments the pointer each time the next-access delay period for a data transfer ends. On completion of the serial transfer that corresponds to the final command comprising the sequence, the SPI sets the pointer to the SPCMD0 register, and in this manner the sequence is executed repeatedly.

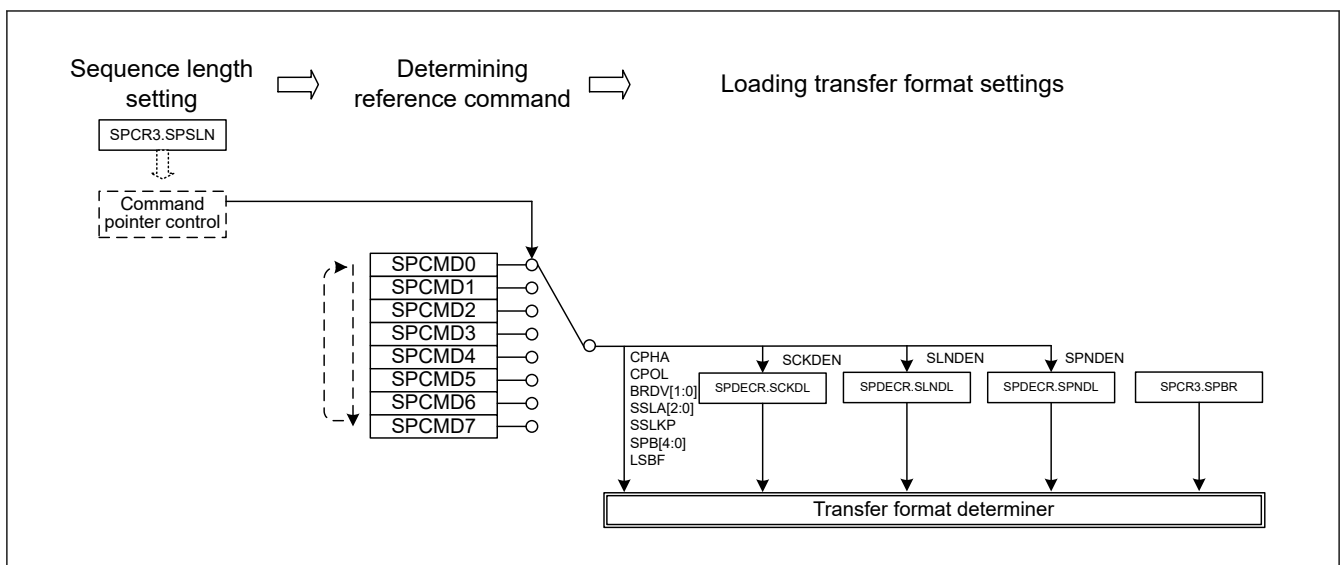
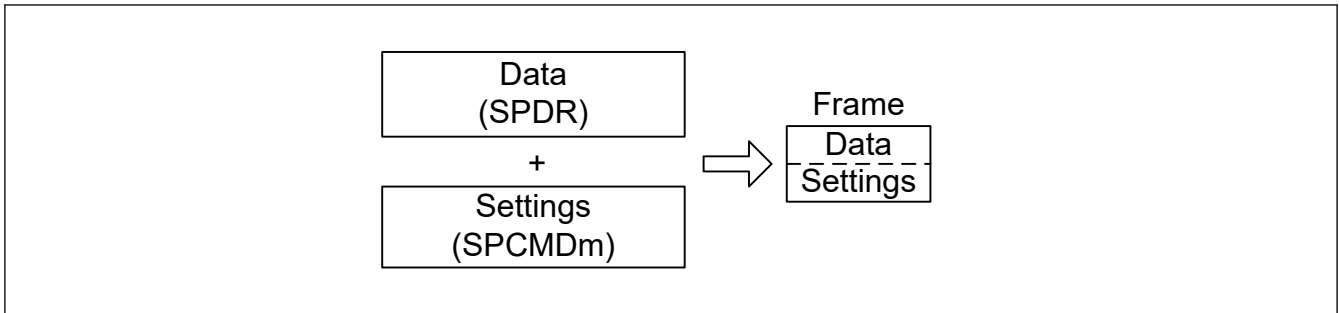


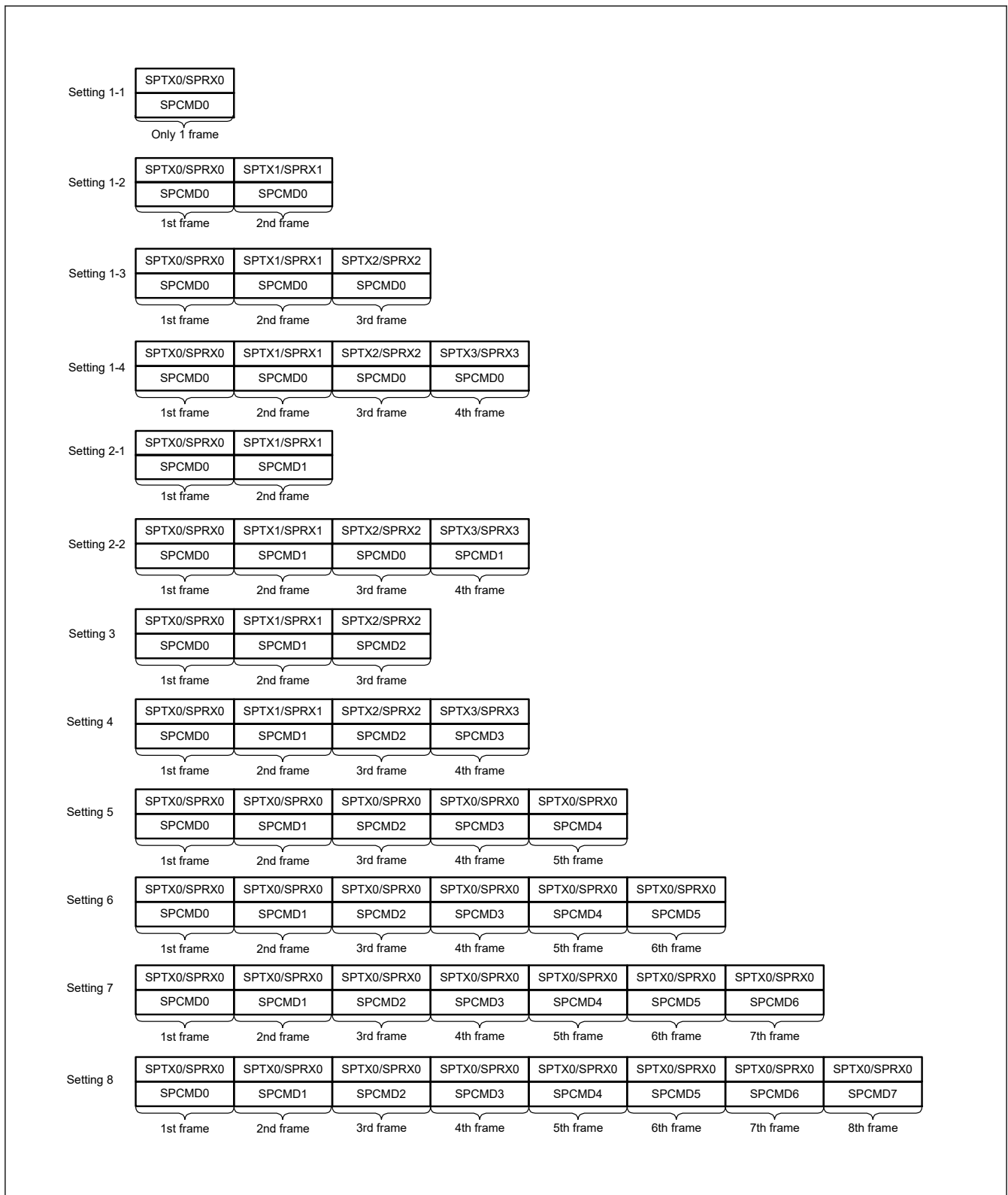
Figure 34.73 Procedure for determining the form of serial transmission in master mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).



**Figure 34.74 Conceptual diagram of frames**

Figure 34.75 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings.

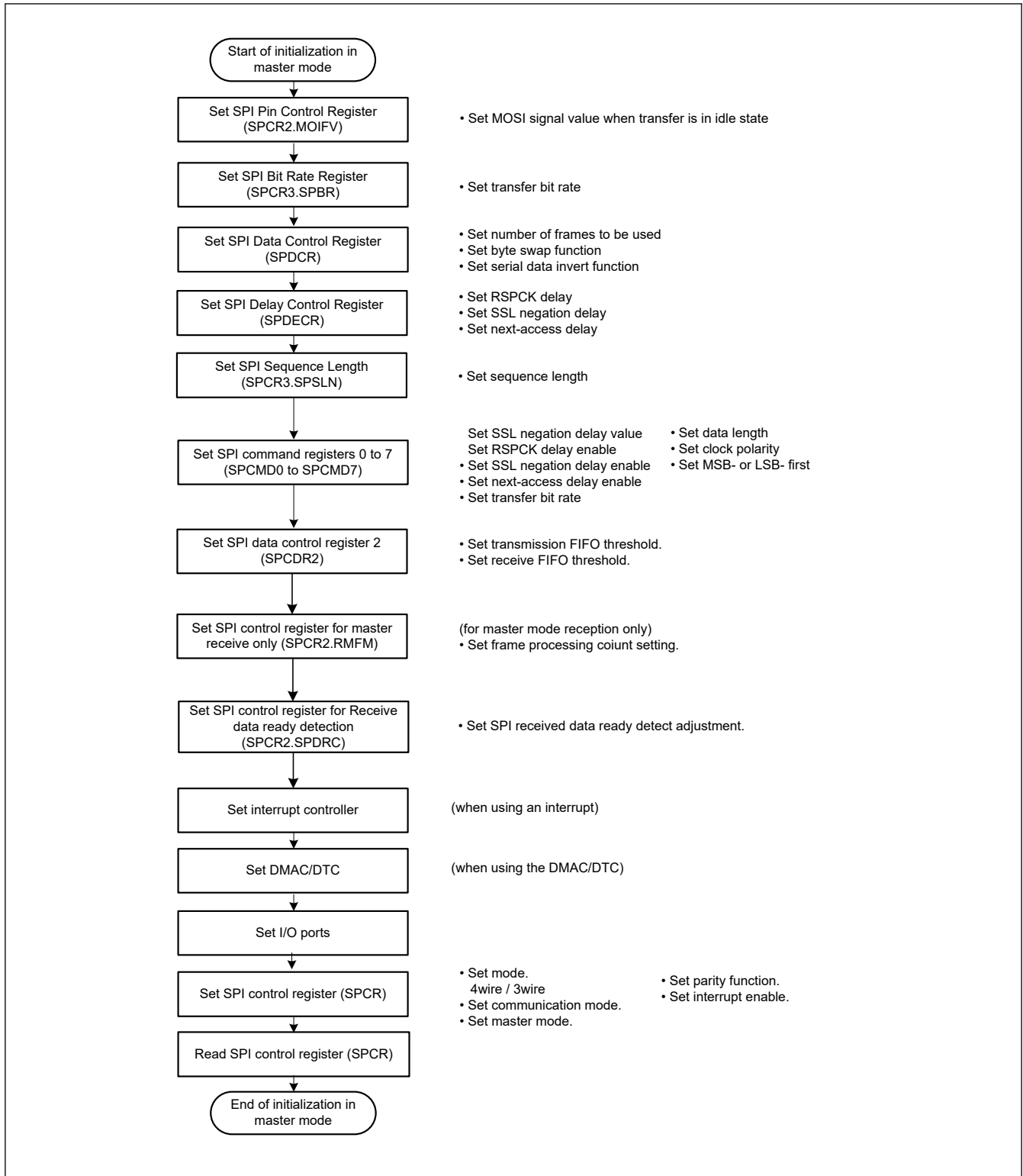


**Figure 34.75 Correspondence between SPI Command Register and transmit and receive buffers in sequence operations**

**(4) Initialization flow**

Figure 34.76 shows an example of initialization flow for clock synchronous operation when the SPI is used in master mode. For information on how to set up the ICU, DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.





**Figure 34.76 Example of initialization flow in master mode for clock synchronous operation**

**(5) Software processing flow**

Software processing during clock synchronous master operation is the same as that for SPI master operation. For details, see (9) Software processing flow in [section 34.3.12.1. Master mode operation](#). Mode fault errors do not occur in clock synchronous operation.

### 34.3.13.2 Slave mode operation

#### (1) Starting serial transfer

When the SPCR.SPMS bit is 1, the first RSPCKn edge triggers the start of a serial transfer in the SPI, and the SPI drives the MISO<sub>n</sub> output signal. The SSL<sub>n0</sub> input signal is not used in clock synchronous operation. For details on the SPI transfer format, see [section 34.3.5. Transfer Formats](#).

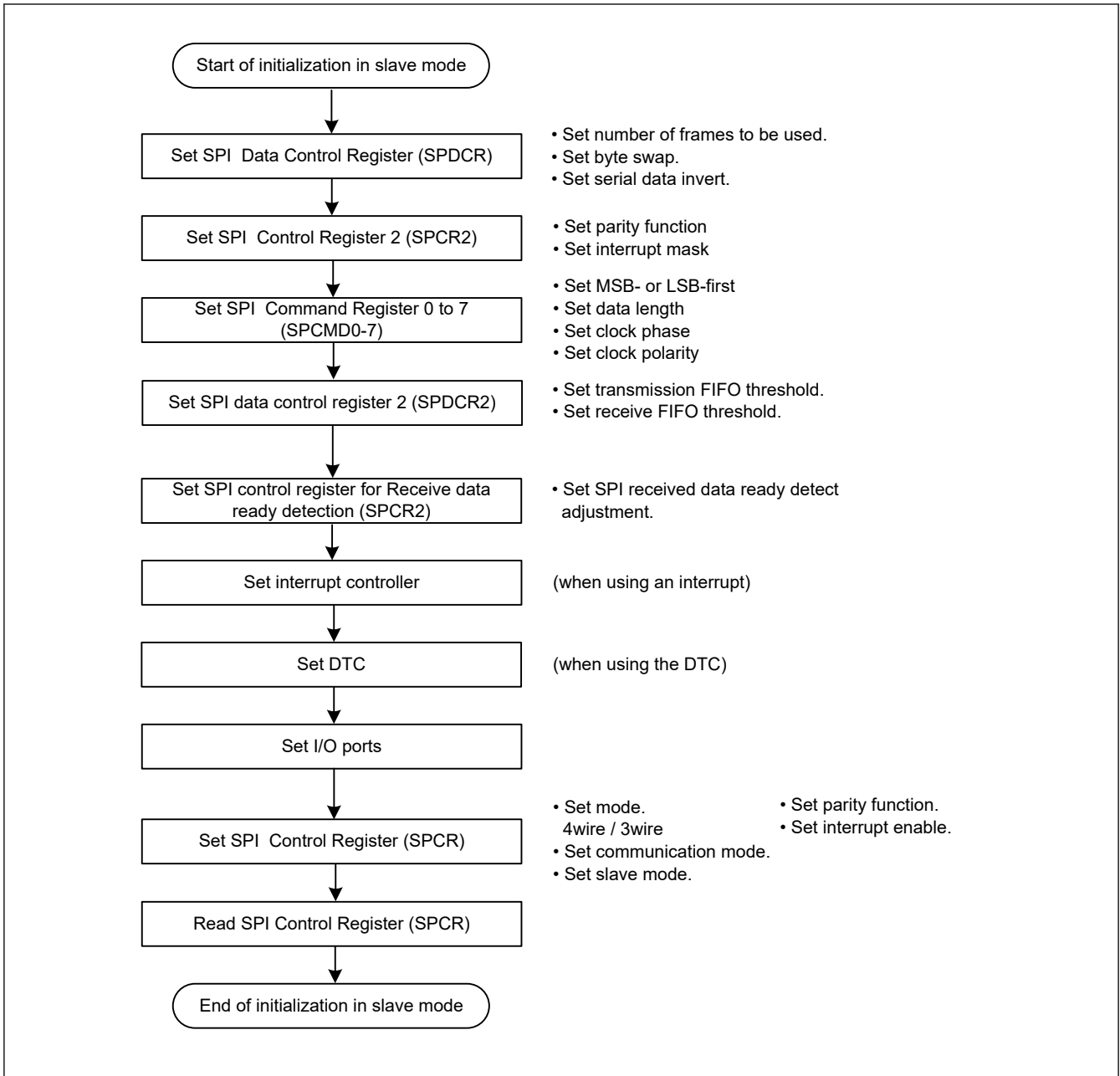
#### (2) Terminating serial transfer

The SPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When the number of data stored in the receive FIFO < the number of FIFO stages, on termination of serial transfer, the SPI copies received data from the shift register to the receive buffer of the SPDR register. On termination of a serial transfer, the SPI changes the status of the shift register to empty regardless of the receive buffer.

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the SPI data length depends on the SPCMD0.SPB[4:0] bits setting. For details on the SPI transfer format, see [section 34.3.5. Transfer Formats](#).

#### (3) Initialization flow

[Figure 34.77](#) shows an example of initialization flow for clock synchronous operation when the SPI is used in slave mode. For a description of how to set up the ICU, DMAC or DTC, and I/O ports, see the descriptions given in the individual blocks.



**Figure 34.77 Example of initialization flow in slave mode for clock synchronous operation**

**(4) Software processing flow**

Software processing during clock synchronous slave operation is the same as that for SPI slave operation. For details, see [\(6\) Software processing flow](#). Mode fault errors do not occur in clock synchronous mode.

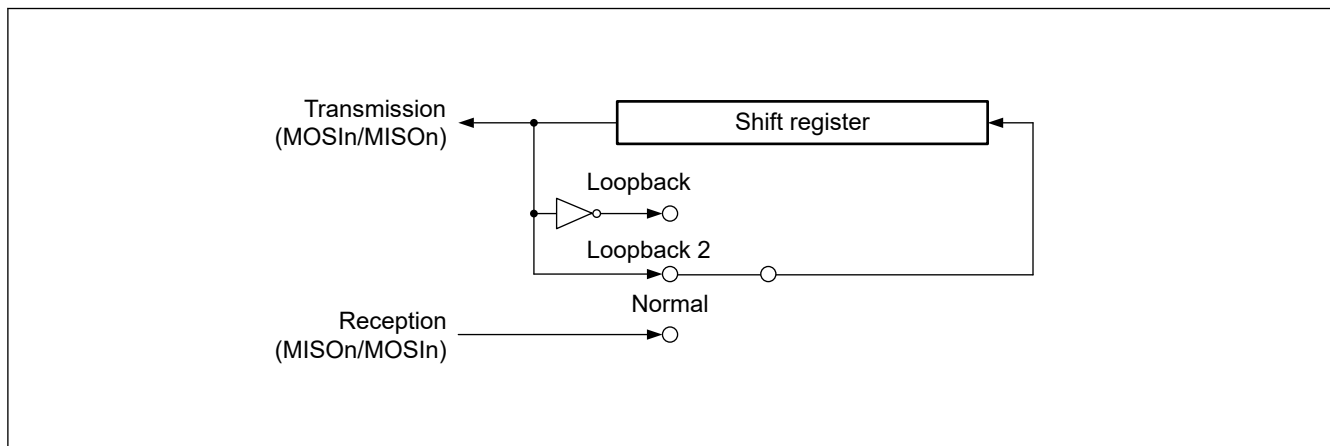
**34.3.14 Loopback Mode**

When 1 is written to the SPCR2.SPLP2 bit or SPCR2.SPLP bit, the SPI shuts off the path between the MISO pin and the shift register if the SPCR.MSTR bit is 1, or between the MOS pin and the shift register if the SPCR.MSTR bit is 0, and connects the input and output paths of the shift register, establishing a loopback mode. The SPI does not shut off the path between the MOS pin and the shift register if the SPCR.MSTR bit is 1, or between the MISO pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the SPI or the reversed transmit data becomes the received data for the SPI.

[Table 34.13](#) lists the relationship between the SPLP2 and SPLP bits and the received data. [Figure 34.78](#) shows the configuration of the shift register I/O paths when the SPI in master mode is set to loopback mode (SPCR2.SPLP2 = 0, SPCR2.SPLP = 1).

**Table 34.13 SPLP2 and SPLP bit settings and received data**

SPCR2.SPLP2 bit	SPCR2.SPLP bit	Received data
0	0	Input data from the MOSIn pin or MISOn pin
0	1	Inverted transmit data
1	0	Transmit data
1	1	Transmit data



**Figure 34.78 Configuration of shift register I/O paths in loopback mode for master mode**

### 34.3.15 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. To detect defects in the parity bit adding unit and error detecting unit, the parity circuit performs self-diagnosis as shown in [Figure 34.79](#).

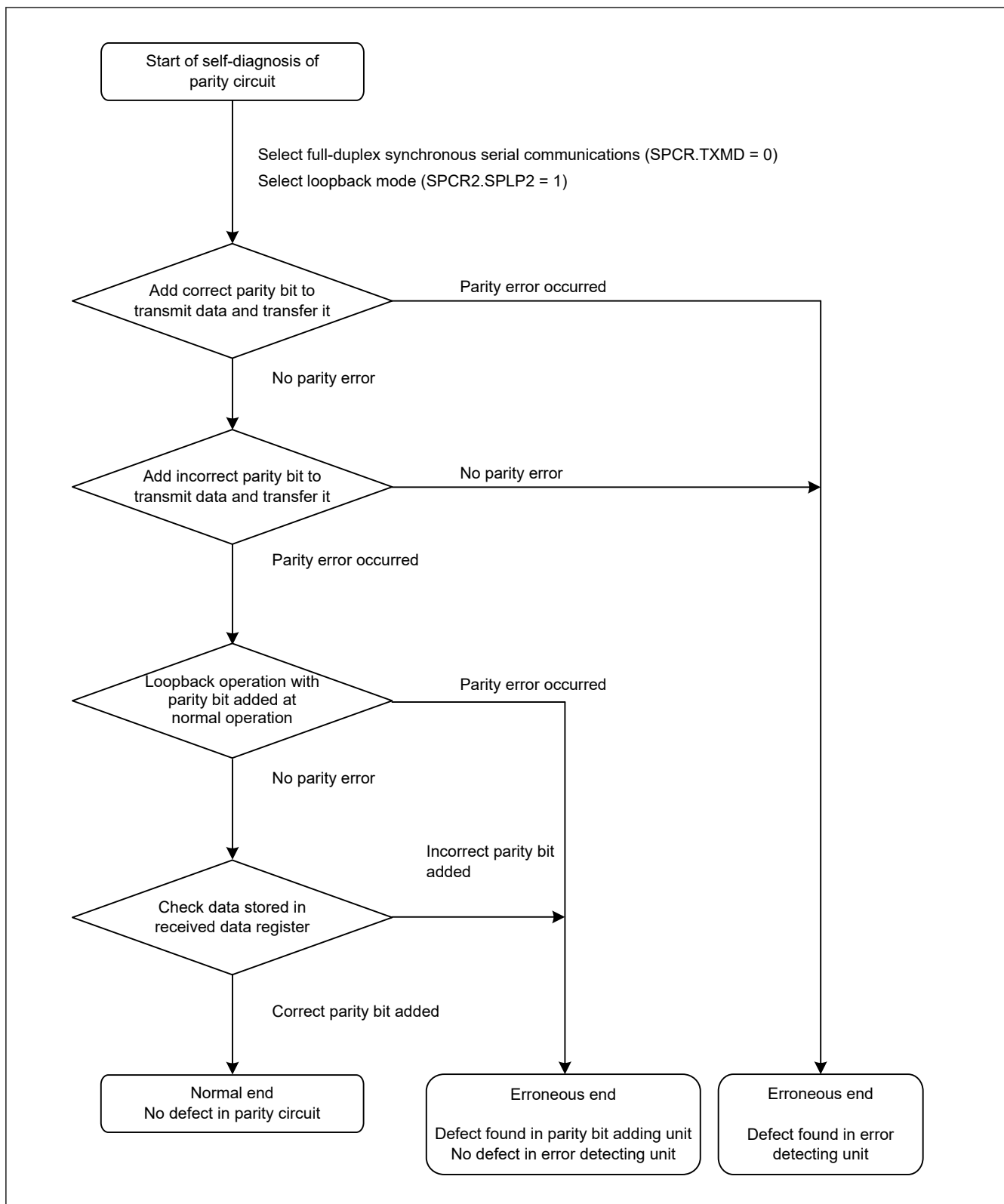


Figure 34.79 Self-diagnosis flow for parity circuit

### 34.3.16 Interrupt Sources

The SPI has the following interrupt sources:

- Receive buffer full
- Transmit buffer empty

- SPI error (mode-fault, underrun, overrun, or parity error)
- SPI idle
- Communication-end

The DMAC or DTC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Because the vector address for the SPIi\_SPEI (SPI error interrupt) is allocated to interrupt requests on mode-fault, underrun, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the SPI are listed in Table 34.14. An interrupt is generated on satisfaction of one of the interrupt conditions in Table 34.14. Clear the receive buffer full and transmit buffer empty sources through a data transfer.

When using the DMAC or DTC to perform data transmission and reception, you must first set up the DMAC or DTC to be in a transfer-enabled status before setting the SPI. For information on setting up the DMAC or DTC, see section 16, [DMA Controller \(DMAC\)](#) and section 17, [Data Transfer Controller \(DTC\)](#).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt occur while the ICU.IELSRn.IR flag is 1, the interrupt is not output as a request for the ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request is output when the ICU.IELSRn.IR flag becomes 0. A retained interrupt request is automatically discarded when it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be set to 0.

**Table 34.14 SPI interrupt sources**

Interrupt source	Symbol	Interrupt condition	DTC/DMAC activation
Receive buffer full	SPIi_SPRI	The receive buffer becomes full (SPSR.SPRF flag is 1) while the SPCR.SPRIE bit is 1 or The receive data become ready (SPSR.SPDRF flag is 1) while the SPCR.SPDRS bit is 0	Possible
Transmit buffer empty	SPIi_SPTI	The transmit buffer becomes empty (SPSR.SPTEF flag is 1) while the SPCR.SPTIE bit is 1	Possible
SPI error (mode-fault, underrun, overrun, or parity error)	SPIi_SPEI	The SPSR.MODF, OVRF, or PERF flag sets to 1, or the SPSR.SPDRS and SPDRF flag set to 1 while the SPCR.SPEIE bit is 1	Impossible
SPI idle	SPIi_SPII	The SPSR.IDLNF flag sets to 0 while the SPCR.SPIIE bit is 1	Impossible
Communication-end	SPIi_SPCEND	CENDIE = 1 and CENDF = 1	Impossible

## 34.4 Event Link Controller Event Output

The Event Link Controller (ELC) can produce the following event output signals:

- Receive buffer full event output
- Transmit buffer empty event output
- Mode-fault, underrun, overrun, or parity error event output
- SPI idle event output
- Transmission-completed event output

The event link output signal is output regardless of the interrupt enable bit setting.

### 34.4.1 Receive Buffer Full Event Output

When the number of data stored in the receive FIFO > the threshold value, or when the number of data stored in the receive FIFO ≤ the threshold value and SPDRES = 0 has elapsed after writing to the receive FIFO, and the SPDRC [7:0] has elapsed outputs an event.

### 34.4.2 Transmit Buffer Empty Event Output

An event is output when the number of empty transmission FIFO stage > the threshold or when the SPCR.SPE bit changes from 0 to 1.

### 34.4.3 Mode-Fault, Underrun, Overrun, Parity Error, or received data ready Event Output

This event signal is output when mode-fault, underrun, overrun, or parity error is detected. See [section 34.5.4. Constraints on Mode-Fault, Underrun, Overrun, Parity Error, or Receive Data Ready Event Output](#) if using this event signal.

#### (1) Mode-fault

[Table 34.15](#) lists the conditions for occurrence of a mode-fault event.

**Table 34.15 Conditions for mode-fault occurrence**

SPI mode	SPCR.MODFEN bit	SSLn0 pin	Remarks
SPI operation (SPMS = 0) Slave (SPCR.MSTR = 0) Motorola-SPI (SPCR.SPFRF = 0)	1	Not active	Event is output only when the SSLn0 pin is deactivated during transmission
SPI operation (SPMS = 0) Slave (SPCR.MSTR = 0) TI-SSP (SPCR.SPFRF = 1)	1	active	Event is output only when the SSLn0 pin is activated during transmission

#### (2) Underrun

This event signal is output in response to an underrun when a serial transfer starts while the transmission data is not ready, and the value of the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1. Under these conditions, the MODF and UDRF flags are set to 1.

#### (3) Overrun

This event signal is output in response to an overrun when a serial transfer completes while the receive buffer contains unread data and the value of the SPCR.TXMD[1:0] bits are 00b or 10b. Under these conditions, the OVRF flag is set to 1.

#### (4) Parity error

This event signal is output in response to a parity error detected on completion of a serial transfer while the value of the SPPE bit in SPCR is 1.

#### (5) Receive Data Ready

When TXMD[1:0] of SPCR = 00b or 10b and SPDRES = 1 as the receive data ready event output condition, the number of data stored in the receive FIFO is received after writing the receive FIFO. An event will be output when the set value of SPDR[7:0] has elapsed while the number is less than the FIFO threshold.

### 34.4.4 SPI Idle Event Output

#### (1) In master mode

In Transmit-Receive / Transmit-only master mode, an event is output when the IDLNF flag in SPSR changes from 1 to 0.

The IDLNF flag changes from 1 to 0 only when either of the conditions 1) and 2) below is met.

- The SPE bit in SPCR is cleared to 0 (SPI initialized) during transmission.
- All of the following three conditions are met.
  - The transmit buffer (SPTXn, n = 0 to 3) is empty (next transfer data has not been set).
  - The SPCP [2:0] bits in SPSR are 000b (at the start of sequence control).
  - Operation completed by the next access delay (when the master main state machine transitions to the idle state).

In receive only master mode

Any of the following 2 conditions is met.

- SPE bit of SPCR is 0 (SPI initialization)
- When any of the following is met
  - When RMFM [4:0] = 0x00, after writing 1 to RMEDTG, operation completed by the next access delay (when the master main state machine transitions to the idle state).

- When RMFM [4:0] ≠ 0x00, after writing 1 to RMEDTG, operation completed by the next access delay (when the master main state machine transitions to the idle state).
- When RMFM [4:0] ≠ 0x00, the SPI internal sequencer transitions to the idle state after operation completed by the next access delay (when the master main state machine transitions to the idle state).

(2) In slave mode

In slave mode, an event is output when the SPCR.SPE bit is set to 0 (SPI is initialized).

34.4.5 Communication End Event Output

In master mode, an event is output when the IDLNF flag (SPI idle flag) changes from 1 to 0. In slave mode, an event occurs with conditions shown in Table 34.16 and Table 34.17

**Table 34.16 Communication End Event Generating Conditions (transmit-receive/transmit slave mode)**

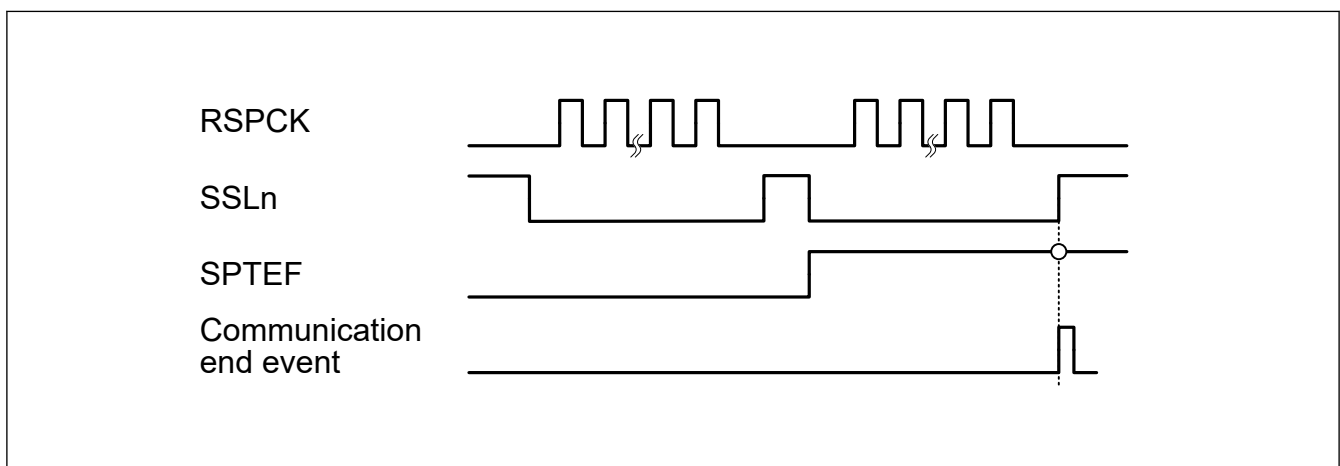
	Transmit Buffer Status	Shift Register Status	Others
SPI operation (SPMS = 0, SPFRF = 0)	Empty	Empty	SSLn0 input is negated
SPI operation (SPMS = 0, SPFRF = 1)	Empty	Empty	SSL negation delay completed
Clock synchronous operation (SPMS = 1)	Empty	Empty	The last even edge of RSPCK of last data was detected (CPHA = 1)

**Table 34.17 Communication End Event Generating Conditions (receive only slave mode)**

	Others
SPI operation (SPMS = 0, SPFRF = 0)	After storing the frames corresponding to the SPFC setting value in the receive buffer, negate SSLn0 input.
SPI operation (SPMS = 0, SPFRF = 1)	After storing the frames corresponding to the SPFC setting value in the receive buffer, SSL negation delay completed
Clock synchronous operation (SPMS = 1)	RSPCK last even edge detection when receiving the last frame for the SPFC set value (CPHA = 1)

Regardless of master mode or slave mode, no event is output when 0 is written to the SPCR.SPE bit during transmission or when the SPCR.SPE bit is cleared due to a mode fault error or an underrun error.

A communication end event is output at the following timing. The communication end event output timing in master operation is omitted because it is output at the same timing as an idle event.



**Figure 34.80 Communication End Event Output Timing (Transmit slave mode, Motorola SPI operation)**



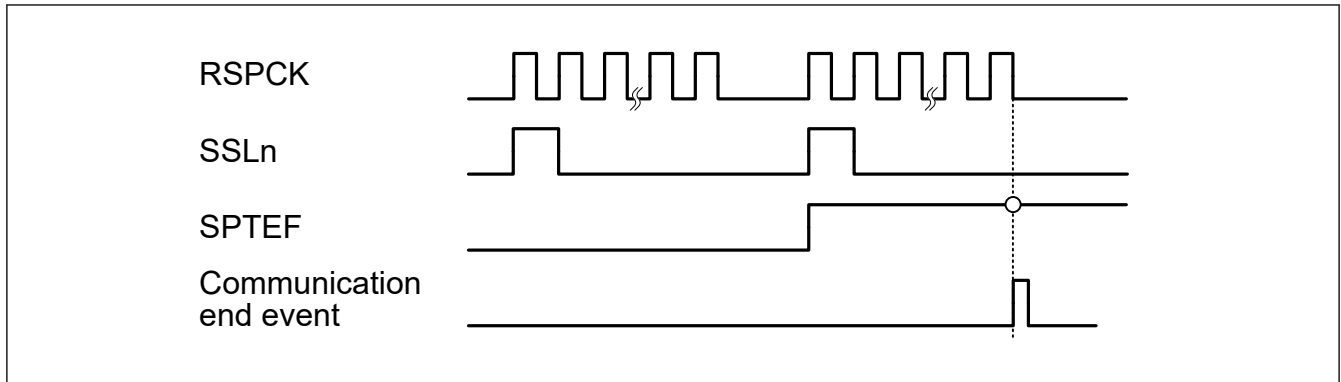


Figure 34.81 Communication End Event Output Timing (Transmit slave mode, TI-SSP Operation)

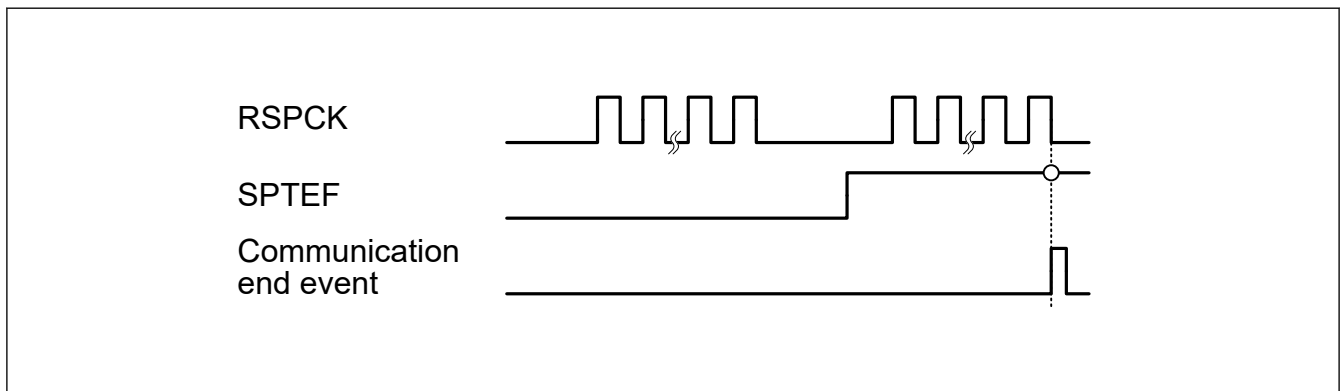


Figure 34.82 Communication End Event Output Timing (Transmit slave mode, Clock Synchronous Operation)

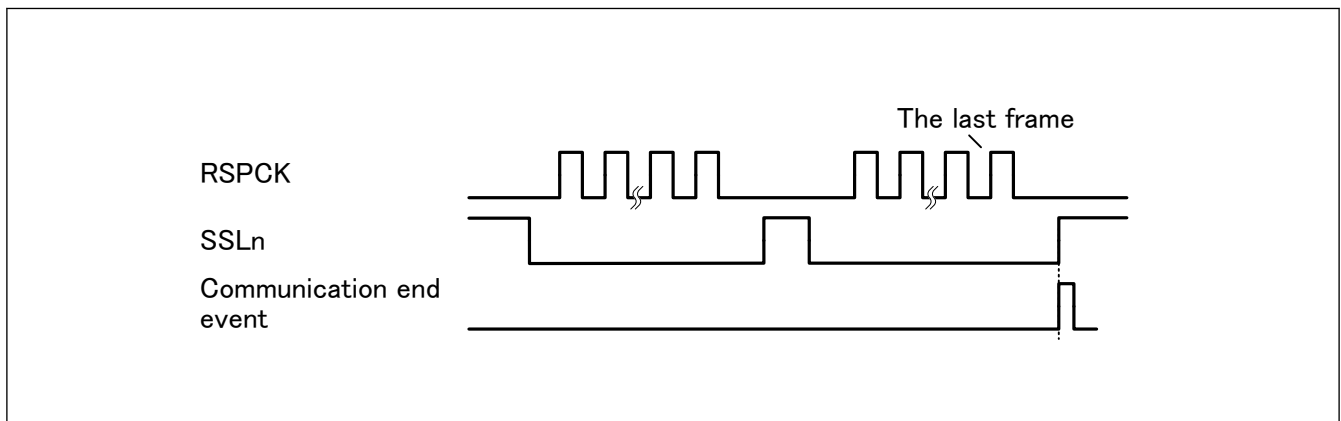


Figure 34.83 Communication End Event Output Timing (Receive only slave mode, Motorola SPI operation)

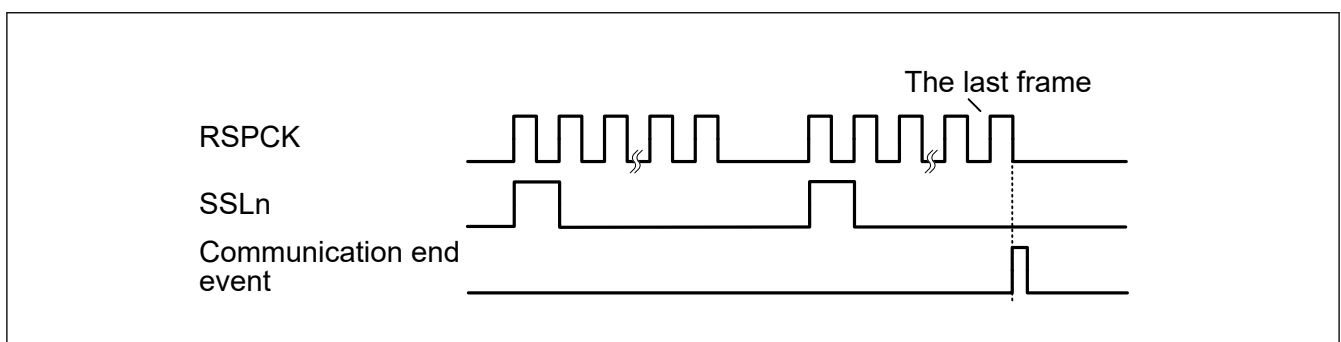
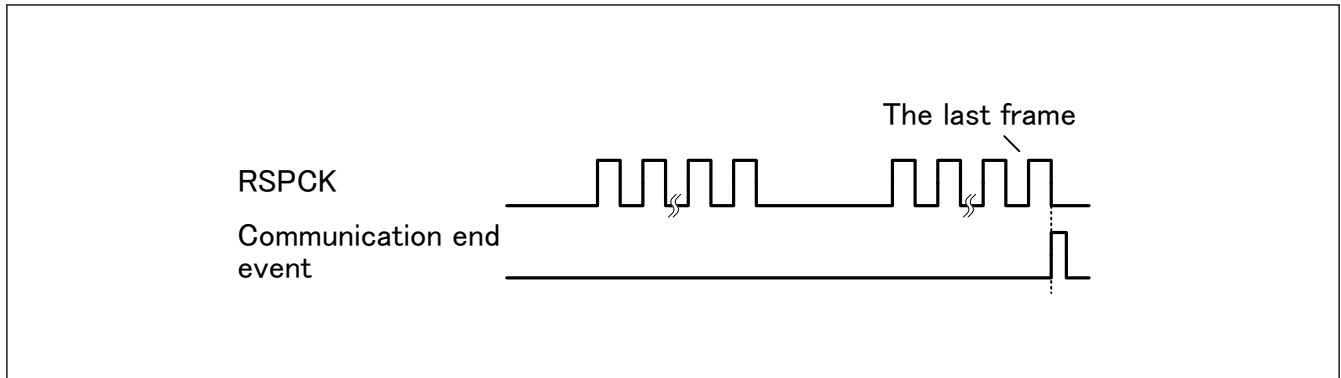


Figure 34.84 Communication End Event Output Timing (Receive only slave mode, TI-SSP Operation)



**Figure 34.85** Communication End Event Output Timing (Receive only slave mode, Clock Synchronous Operation)

### 34.4.6 Synchronization bypass function

SPI has an internal clock (PCLK) and an operation clock (TCLK), and each has its own operation circuit. Therefore, a synchronization circuit is inserted between the signals between different clocks, and a signal delay between different clocks requires a synchronization delay time.

However, the synchronization circuit can be bypassed by the BPEN = 1 of the SPI control register (SPCR) only when the same clock is input as the internal bus clock and the operation clock. In this case, the synchronization delay time is excluded, and responsiveness is improved.

In addition, SPI has a synchronization circuit between the communication clock (RSPCK) and the operation clock (TCLK), but this synchronization circuit cannot be bypassed.

## 34.5 Usage Notes

### 34.5.1 Settings for the Module-Stop State

The Module Stop Control Register B (MSTPCRB) can enable or disable the SPI operation. The SPI is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details on the Module Stop Control Register B, see [section 10, Low Power Modes](#).

### 34.5.2 Constraint on Low-Power Functions

When using the module-stop function and entering a low-power mode other than CPU Sleep mode or CPU Deep Sleep mode, set the SPCR.SPE bit to 0 before completing communication.

### 34.5.3 Constraints on Starting Transfer

If the ICU.IELSRn.IR flag is 1 when transfer starts, the interrupt request is internally retained, which can lead to unanticipated behavior of the ICU.IELSRn.IR flag.

To prevent this, use the following procedure to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1):

1. Confirm that transfer stopped (the SPCR.SPE bit is 0).
2. Set the associated interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) to 0.
3. Read the associated interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) and confirm that its value is 0.
4. Set the ICU.IELSRn.IR flag to 0.

### 34.5.4 Constraints on Mode-Fault, Underrun, Overrun, Parity Error, or Receive Data Ready Event Output

Using the mode-fault, underrun, overrun, parity error, or receive data ready event is prohibited if the SPI is in multi-master mode (when the SPCR.SPMS bit is 0, the SPCR.MSTR bit is 1, and the SPCR.MODFEN bit is 1).

### 34.5.5 Constraints on the SPSR.SPRF and SPSR.SPTEF Flags

If the polling flags, SPRF and SPTEF, are used, using the interrupts is prohibited, and you must set the SPCR.SPRIE and SPCR.SPTIE bits to 0. Either the interrupts or the flags can be used, but not both.

## 35. Octal Serial Peripheral Interface (OSPI)

This is the OSPI\_B version of the OSPI peripheral module.

OSPI\_B is referred to as OSPI in this chapter.

### 35.1 Overview

The xSPI (eXpanded Serial Peripheral Interface) protocol specifies the interface for Non-Volatile Memory Devices, which provides high data throughput, low signal count, and limited backward compatibility with legacy SPI devices. The electrical interface can deliver up to 200 Mbytes per second raw data throughput. The OSPI is compliant with JEDEC standard JESD251 (Profile 1.0 and 2.0), JESD251-1 and JESD252.

JESD251 specifies two interface profiles where profile 1.0 is Octal SPI and profile 2.0 is HyperBus™ (HyperRAM™ and HyperFlash™).

OSPI supports QSPI protocol.

Table 35.1 lists the OSPI specifications, Figure 35.1 shows a block diagram, and Table 35.2 lists the I/O pins.

**Table 35.1 OSPI specifications**

Item	Description
Protocol	Compliant with the xSPI protocol
Data transmission and reception	Issue the transaction for up to 2 Slave as Master Only one of the memory devices can operate at a time.
Transfer speed	Support the transfer at xSPI200
Mode	<ul style="list-style-type: none"> <li>• Support Protocol modes below               <ul style="list-style-type: none"> <li>– 1/4/8pin with SDR/DDR (1S-1S-1S, 4S-4D-4D, 8D-8D-8D)</li> <li>– 2/4pin with SDR (1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S)</li> </ul> </li> <li>• Configurable address length</li> <li>• Configurable initial access latency cycle</li> <li>• Support XiP mode</li> </ul>
OSPI function	<ul style="list-style-type: none"> <li>• Support Write Data Mask</li> <li>• Support In-band Reset</li> <li>• Memory-mapping               <ul style="list-style-type: none"> <li>– Support up to 256 MB address space each CS</li> <li>– Prefetch function for burst-read with low latency</li> <li>– Outstanding buffer for burst-write with high throughput</li> </ul> </li> <li>• Manual command               <ul style="list-style-type: none"> <li>– Configurable up to 4 commands</li> <li>– Status Register Polling function</li> </ul> </li> <li>• Input Strobe port timing shift</li> </ul>
Transfer target	Each bus master More details, see <a href="#">section 14, Buses</a>
Interrupt source	Error interrupt Completion interrupt
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security attribution can be set for IO register area External address space is defined as Non-secure

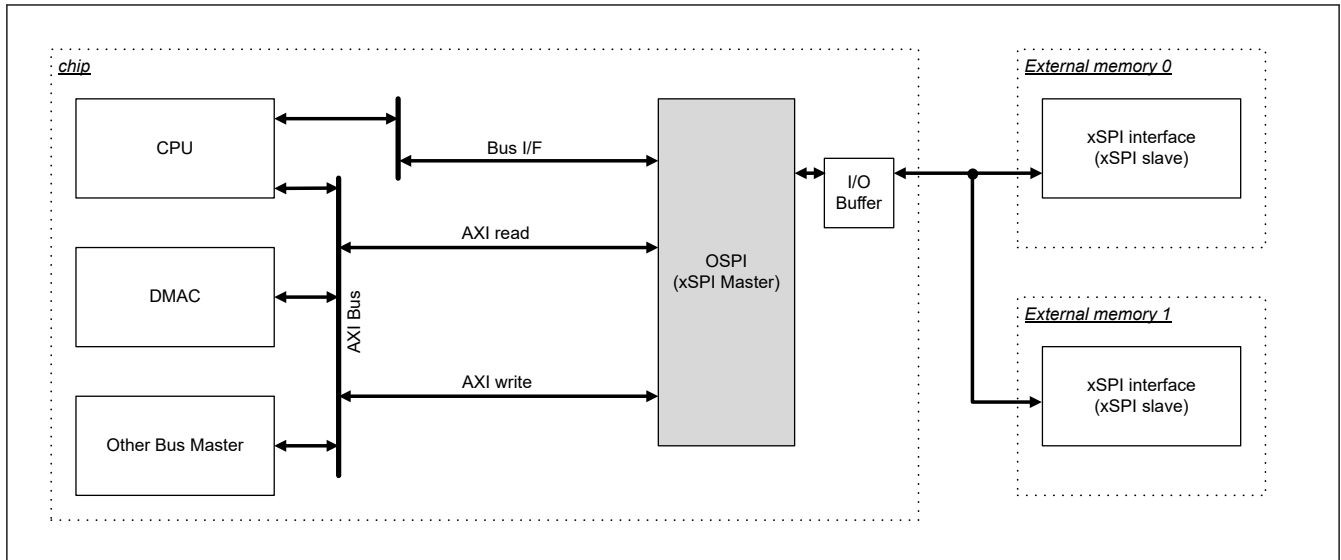


Figure 35.1 Block diagram

Table 35.2 OSPI I/O pins

Pin name	I/O	Function
OM_SCLK	Output	Clock Positive
OM_SCLKN	Output	Clock Negative
OM_CS0	Output	Chip Select for slave0
OM_CS1	Output	Chip Select for slave1
OM_DQS	I/O	Read Data Strobe / Write Data Mask
OM_SIO0	I/O	Data 0 input/output
OM_SIO1	I/O	Data 1 input/output
OM_SIO2	I/O	Data 2 input/output
OM_SIO3	I/O	Data 3 input/output
OM_SIO4	I/O	Data 4 input/output
OM_SIO5	I/O	Data 5 input/output
OM_SIO6	I/O	Data 6 input/output
OM_SIO7	I/O	Data 7 input/output
OM_RESET	Output	Master reset status for slave0,1
OM_RSTO1	Input	Slave reset status for slave1
OM_ECSINT1	Input	Interrupt for slave1 / Error Correction Status for slave1
OM_WP1	Output	Write Protect for slave1

Note: For OM\_SIO7-0, OM\_SCLK, OM\_SCKN and OM\_DQS pins, 36Ω±5% resistor need to be put on the board to comply with JESD251 I/O driver definition. It is recommended for the operation with proper signal quality.

## 35.2 Register Descriptions

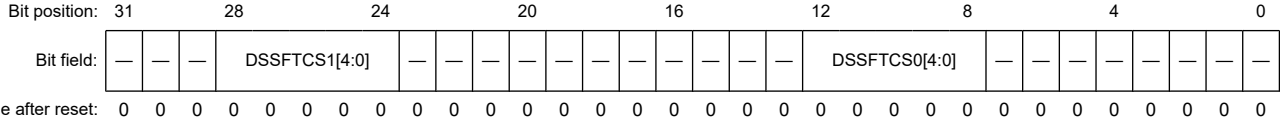
### 35.2.1 OSPI Configuration Registers

These registers configure xSPI Master function. These registers should be configured in the initialization phase before issuing xSPI transaction.

### 35.2.1.1 WRAPCFG : OSPI Wrapper Configuration Register

Base address: OSPI0\_B = 0x4026\_8000  
 OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x000



Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DSSFTCS0[4:0]	OM_DQS shift for slave0 This field configures the number of delay cell for a OM_DQS port. It is used to adjust the OM_DQS sampling timing. When automatic calibration is enabled, it could be updated automatically. In this case, it shall not be written by user. 0x00: No shift 0x01: Add a delay of 1 cell ⋮ 0x1E: Add a delay of 30 cells 0x1F: Add a delay of 31 cells	R/W
23:13	—	These bits are read as 0. The write value should be 0.	R/W
28:24	DSSFTCS1[4:0]	OM_DQS shift for slave1 The function is same as one of slave0.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

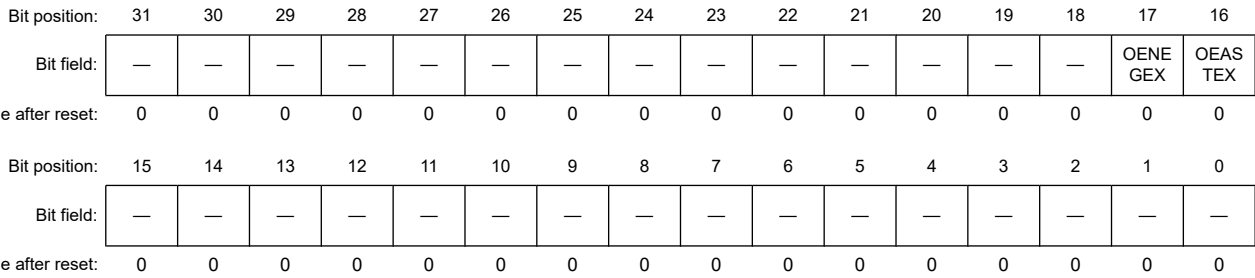
Note: S-TYPE-3, P-TYPE-3

This register has functions to configure xSPI Master function.

### 35.2.1.2 COMCFG : OSPI Common Configuration Register

Base address: OSPI0\_B = 0x4026\_8000  
 OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x004



Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	OEASTEX	Output Enable Asserting extension This bit extends 1 cycle output enable of OM_SIO7-0 and OM_DQS during output enable asserting. When set to 1, OM_CS <sub>n</sub> (n = 0, 1) asserting should be extended (LIOCFCGS <sub>n</sub> .CSASTEX = 1 (n = 0, 1)). This bit shall not be used in case of no latency cycle. Because OSPI output data could be conflicted with OSPI input data. 0: No extend 1 cycle Output enable 1: Extend 1 cycle Output enable	R/W

Bit	Symbol	Function	R/W
17	OENEGEX	Output Enable Negating extension This bit extends 1 cycle output enable of OM_SIO7-0 and OM_DQS during output enable negating. This bit should not be used in case of no latency cycle. Because OSPI output data could be conflicted with OSPI input data. 0: No extend 1 cycle Output enable 1: Extend 1 cycle Output enable	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to configure xSPI Master function.

### 35.2.1.3 BMCFGCHn : OSPI Bridge Map Configuration Register chn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x008 + 0x004 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16							
Bit field:	CMBTIM[7:0]														—	—	—	—	—	—	—	—	PREEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0							
Bit field:	MWRSIZE[7:0]														MWR COMB	—	—	—	—	—	—	—	WRMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						

Bit	Symbol	Function	R/W
0	WRMD	System bus Write Response mode This bit selects the timing of System bus write response in memory-mapping mode. When set to 1, it returns the response after transmitting a frame on xSPI bus. When enabled this mode, Memory Write Combination mode must be disabled. 0: Return response after storing to Internal Write Buffer 1: Return response after issuing write transaction to xSPI bus	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	MWRCOMB	Memory Write Combination mode This bit selects to combine the OSPI data in write access of memory-mapping mode. When set to 0, OSPI data size depends on system bus's burst type and size. When this field is set to "1", the data size depends on MWRSIZE[7:0] field.*1 When set to 1, any write transaction could be held in this xSPI master temporarily. 0: Disable combination mode 1: Enable combination mode	R/W
15:8	MWRSIZE[7:0]	Memory Write Size These bits select the size to combine incremental address in memory-mapping mode. It transmits an xSPI frame with the data combined up to the configured size while the address is incremental. When detected non-incremental address or a read transaction before reaching to the target size, it transmits the pending data into xSPI bus. 0x00: Combine incremental address up to 4 bytes 0x01: Combine incremental address up to 8 bytes ⋮ 0x0E: Combine incremental address up to 60 bytes 0x0F: Combine incremental address up to 64 bytes 0xFF: Combine incremental address up to 2 bytes Others: Setting prohibited	R/W
16	PREEN	Prefetch enable This bit enables prefetch function for read transaction in memory-mapping mode. It could reduce the latency for read transaction with incremental address. 0: Disable prefetch function 1: Enable prefetch function	R/W

Bit	Symbol	Function	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
31:24	CMBTIM[7:0]	Combination timer This field specifies expiration period of combination timer. This timer is counted by PCLKA. 0x00 means disabling the combination timer. When the timer is expired, the data in the combination buffer is pushed to memory device.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Writing to OSPI memory space other than 64 bit is prohibited during combination mode.

This register has functions to configure xSPI Master function.

### 35.2.1.4 CMCFG0CSn : OSPI Command Map Configuration Register 0 CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x010 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ADDRPCD[7:0]								ADDRPEN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	ARYA MD	WPBS TMD	ADDSIZE[1:0]	FFMT[1:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	FFMT[1:0]	Frame format These bits configure xSPI frame format in memory-mapping mode. Please see <a href="#">Table 35.7</a> for detail. 0 0: Normal format: Command 1 byte, Address ADDSIZE, Data up to system bus transaction. 0 1: 8D-8D-8D profile 1.0 format: Command 2 bytes, Address ADDSIZE, Data up to system bus transaction 1 0: 8D-8D-8D profile 2.0 Command Modifier format: Command & Modifier 6 bytes, Data up to system bus transaction 1 1: 8D-8D-8D profile 2.0 Commands with Extended Command Modifier format: Command & Modifier 6 bytes, Data up to system bus transaction	R/W
3:2	ADDSIZE[1:0]	Address size These bits configure the number of address byte in memory-mapping mode. In case of 8D-8D-8D profile 2.0, it should be configured to 4 bytes. 0 0: 1 byte (256-byte address space) 0 1: 2 bytes (64 KB address space) 1 0: 3 bytes (16 MB address space) 1 1: 4 bytes (4 GB address space)	R/W
4	WPBSTMD	Wrapping burst mode When this field is set to 1, the wrapping boundary between system bus access and xSPI memory shall be mached. 0: Separate xSPI transfer at the wrapping address boundary 1: Not separate xSPI transfer at the wrapping address boundary	R/W
5	ARYAMD	Array address mode When this field is set to 1, address for memory is mapped as {A[25:10], A[9:4], 6{RSV}, A[3:0]} where A[25:0] is normal address, and RSV is reserved value(0b). This field is effective only when FFMT=1d. 0: Normal address mode 1: Array address mode	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W



Bit	Symbol	Function	R/W
23:16	ADDRPEN[7:0]	Address Replace Enable These bits select the bits to replace for MSByte of System bus address in memory-mapping mode. 0: No replacement (xSPI frame address field is same as System bus address) 1: Replacement	R/W
31:24	ADDRPCD[7:0]	Address Replace Code These bits configure the code to replace the MSByte of System bus address in memory-mapping mode. It replaces the corresponding bits when Address Replace Enable bit is set to 1.	R/W

Note: S-TYPE-3, P-TYPE-3

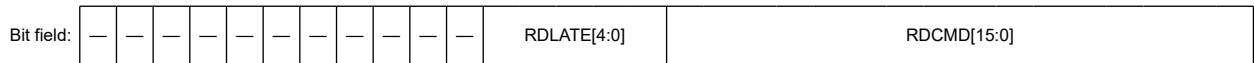
This register has functions to configure xSPI Master function.

### 35.2.1.5 CMCFG1CSn : OSPI Command Map Configuration Register 1 CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x014 + 0x010 × n

Bit position: 31 20 16 15 0



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

Bit	Symbol	Function	R/W
15:0	RDCMD[15:0]	Read command These bits configure the command field of read transaction in memory-mapping mode. Normal format and 8D-8D-8D profile 2.0 format use only upper 1 byte. 8D-8D-8D profile 1.0 format uses 2 bytes.	R/W
20:16	RDLATE[4:0]	Read latency cycle These bits configure the latency cycle of read transaction in memory-mapping mode. 0x00: No latency 0x01: 1 cycle ⋮ 0x08: 8 cycle (default) ⋮ 0x1E: 30 cycles 0x1F: 31 cycles Others: Setting prohibited	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

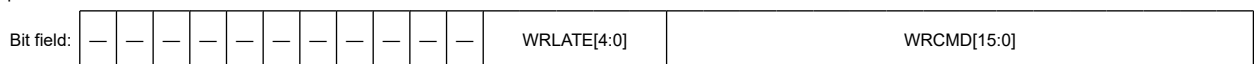
This register has functions to configure xSPI Master function.

### 35.2.1.6 CMCFG2CSn : OSPI Command Map Configuration Register 2 CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x018 + 0x010 × n

Bit position: 31 20 16 15 0



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

Bit	Symbol	Function	R/W
15:0	WRCMD[15:0]	Write command These bits configure the command field of write transaction in memory-mapping mode. Normal format and 8D-8D-8D profile 2.0 format use only upper 1 byte. 8D-8D-8D profile 1.0 format uses 2 bytes.	R/W
20:16	WRLATE[4:0]	Write latency cycle These bits configure the latency cycle of write transaction in memory-mapping mode. 0x00: No latency 0x01: 1 cycle ⋮ 0x08: 8 cycle (default) ⋮ 0x1E: 30 cycles 0x1F: 31 cycles Others: Setting prohibited	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to configure xSPI Master function.

### 35.2.1.7 LIOCFGCSn : OSPI Link I/O Configuration Register CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x050 + 0x004 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DDRSMPLEX[3:0]				SDRSMPSTFT[3:0]				SDRS MPMD	SDRD RV	CSNE GEX	CSAS TEX	CSMIN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WRMS KMD	LATE MD	PRTMD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PRTMD[9:0]	Protocol mode These bits configure the protocol mode and the pin to sample data inputs. In case of using not SPI clock but Data strobe for sampling in SDR mode, it is required to set PRTMD[9] to 1. 0x000: 1S-1S-1S 0x3B2: 4S-4D-4D 0x3FF: 8D-8D-8D 0x048: 1S-2S-2S 0x049: 2S-2S-2S 0x090: 1S-4S-4S 0x092: 4S-4S-4S Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W													
10	LATEMD	<p>Latency mode This bit selects the behavior of initial access latency phase for both direct-manual mode and memory-mapping mode. When set to 0, the latency cycle is equal to each configured cycle from transmitting address field. When set to 1, the latency cycle is incremented from the last byte-pair of Address field and is extended 2 times of each configured cycle depending on OM_DQS port. It is used only for profile 2.0 frame format of 8D-8D-8D protocol mode with 6 bytes command/address field. Please refer to xSPI protocol for detail.</p> <table border="1"> <thead> <tr> <th>Value: Function</th> <th>Frame format</th> <th>Usage</th> </tr> </thead> <tbody> <tr> <td rowspan="2">0: Configurable latency</td> <td>profile 2.0</td> <td>The configurable latency cycle should be set as minus 1.</td> </tr> <tr> <td>Others</td> <td>Latency cycle increments after address field.</td> </tr> <tr> <td rowspan="2">1: Variable latency</td> <td>profile 2.0</td> <td>Latency cycle increments from address [23:16]. And it should not be set to 1.</td> </tr> <tr> <td>Others</td> <td>Not support</td> </tr> </tbody> </table>	Value: Function	Frame format	Usage	0: Configurable latency	profile 2.0	The configurable latency cycle should be set as minus 1.	Others	Latency cycle increments after address field.	1: Variable latency	profile 2.0	Latency cycle increments from address [23:16]. And it should not be set to 1.	Others	Not support	R/W
Value: Function	Frame format	Usage														
0: Configurable latency	profile 2.0	The configurable latency cycle should be set as minus 1.														
	Others	Latency cycle increments after address field.														
1: Variable latency	profile 2.0	Latency cycle increments from address [23:16]. And it should not be set to 1.														
	Others	Not support														
11	WRMSKMD	<p>Write mask mode This bit selects to use OM_DQS port as write data mask. It could be useful for write access of odd byte. It is used only for 8D-8D-8D protocol mode. 0: Write mask disable 1: Write mask enable</p>	R/W													
15:12	—	These bits are read as 0. The write value should be 0.	R/W													
19:16	CSMIN[3:0]	<p>CS minimum idle term This bit configures the minimum cycle between xSPI frames. 0x0: 1 cycle 0x1: 2 cycles ⋮ 0x7: 8 cycles (default) ⋮ 0xE: 15 cycles 0xF: 16 cycles Others: Setting prohibited</p>	R/W													
20	CSASTEX	<p>CS asserting extension This bit extends 1 cycle chip select pins when asserting. 0: No extension 1: Extend 1 cycle</p>	R/W													
21	CSNEGEX	<p>CS negating extension This bit extends 1 cycle chip select pins when negating. 0: No extension 1: Extend 1 cycle</p>	R/W													
22	SDRDRV	<p>SDR driving timing This bit configures the timing of data output in SDR. This bit should not be set to 1 in case of no latency cycle. Because OSPI output data could be conflicted with OSPI input data. 0: Drive at 1/2 cycle before CK rising-edge 1: Drive at CK rising-edge</p>	R/W													
23	SDRSMPMD	<p>SDR Sampling mode This bit selects the edge of sampling in SDR. In DDR, regardless of this setting, it samples OM_SIO<sub>n</sub> (n = 0 to 7) ports with both edges of OM_DQS. When set to 1, it samples at rising-edge before falling-edge. 0: Samples data input at falling-edge 1: Samples data input at rising-edge</p>	R/W													

Bit	Symbol	Function	R/W
27:24	SDRSMPST[3:0]	SDR Sampling window shift These bits shift the timing of CK sampling in SDR. In case of using OM_DQS in SDR, there is no influence on the behavior. In case of DDR or using OM_DQS in SDR, it should be set to 0.  0x0: Sample without delay 0x1: Sample at 1 cycle delay ⋮ 0x6: Sample at 6 cycle delay 0x7: Sample at 7 cycle delay Others: Setting prohibited	R/W
31:28	DDRSMPST[3:0]	DDR sampling window extend These bits configure the cycle of extending the sampling window in DDR. In DDR, the input data is sampled during the expected cycle soon after latency cycle. The input data out of range is ignored. It can be configured depending on OM_DQS propagation delay.  0x0: Expand no cycle 0x1: Expand 1 cycle ⋮ 0x6: Expand 6 cycles 0x7: Expand 7 cycles Others: Setting prohibited	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to configure xSPI Master function.

### 35.2.2 OSPI Control Registers

These registers control xSPI Master function.

#### 35.2.2.1 BMCTL0 : OSPI Bridge Map Control Register 0

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CH0CS1ACC[1:0]	CH0CS0ACC[1:0]	—	—
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	CH0CS0ACC[1:0]	System bus ch0 to slave0 memory area access enable This field enables the access from ch0 to CS0 memory.  0 0: Read/Write disable 0 1: Read enable, Write disable 1 0: Read disable, Write enable 1 1: Read/Write enable	R/W
3:2	CH0CS1ACC[1:0]	System bus ch0 to slave1 memory area access enable This field enables the access from ch0 to CS1 memory.  0 0: Read/Write disable 0 1: Read enable, Write disable 1 0: Read disable, Write enable 1 1: Read/Write enable	R/W
7:4	—	These bits are read as 1. The write value should be 1.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

This register shall be configured in the initialization phase. When setting is needed to be changed after beginning xSPI transaction, please stop all communication, see [section 35.3.7.2. Flow of Communication Stop](#) before changing the value of BMCTL0.

### 35.2.2.2 BMCTL1 : OSPI Bridge Map Control Register 1

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PBUF CLRC H0	—	MWRP USHC H0	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	The write value should be 0.	W
8	MWRPUSHCH0	Memory Write Data Push for ch0 This field requests to push the pending data in combination mode.*1 0: No command 1: Push request	W
9	—	The write value should be 0.	W
10	PBUFCLRCH0	Prefetch Buffer clear for ch0 This field requests to clear the prefetch buffer when the prefetch function is enabled.*1 It should not be set during memory access (COMSTT.MEMACCCH0 = 1). 0: No command 1: Clear request	W
31:11	—	The write value should be 0.	W

Note: S-TYPE-3, P-TYPE-3

Note 1. Prefetch/Combination behavior is not defined when cycle base race condition between asserting these bits and system bus access is occurred.

This register has functions to control xSPI Master function.

### 35.2.2.3 CMCTLCHn : OSPI Command Map Control register chn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x0068 + 0x004 × n (n = 0, 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	XIPEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	XIPEXCODE[7:0]								XIPENCODE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	XIPENCODE[7:0]	XiP mode enter code These bits configure the code to enter XiP mode in memory-mapping mode.	R/W
15:8	XIPEXCODE[7:0]	XiP mode exit code These bits configure the code to exit XiP mode in memory-mapping mode.	R/W
16	XIPEN	XiP mode enable This bit enables XiP mode in memory-mapping mode. When set to 1, XiP enter code is inserted in the latency field, and the command field in next transaction is omitted. When set to 0, XiP exit code is inserted in the latency field. And it is set to 0 automatically when transmitting XiP disable pattern. It should not be used for 8D-8D-8D protocol mode profile 2.0 frame format. 0: Disable XiP mode 1: Enable XiP mode	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

### 35.2.2.4 CDCTL0 : OSPI Command Manual Control Register 0

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	PERREP[3:0]				—	—	—	PERITV[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TRNUM[1:0]	CSSE L	—	PERM D	TRRE Q	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRREQ	Transaction request This bit requests to issue the transaction of manual-command. When set to 1, it starts the transaction. It is cleared to 0 when the transaction completed. The transaction is canceled by clearing to 0 while the transaction is ongoing. 0: No transaction 1: Request transaction	R/W
1	PERMD	Periodic mode This bit enables the periodic transaction mode. When set to 1, it repeats a transaction periodically and compares the read value with the expected value. It alternates the status polling operation for external memory. 0: Direct manual-command mode 1: Periodic manual-command mode	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue manual-command. 0: CS0 1: CS1	R/W
5:4	TRNUM[1:0]	Transaction number These bits configure the number of transactions in normal manual-command mode. In periodic manual-command, regardless of this setting, the read data of last command is compared. 0 0: Issue 1 command (using command buffer 0) 0 1: Issue 2 commands (using command buffer 0-1) 1 0: Issue 3 commands (using command buffer 0-2) 1 1: Issue 4 commands (using command buffer 0-3)	R/W

Bit	Symbol	Function	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
20:16	PERITV[4:0]	Periodic transaction interval These bits configure the interval of transaction in periodic manual-command mode. Too short interval compared with CPU bus cycle could result in no store into command buffer0. The interval should be longer than 4 times of CPU bus cycle. 0x00: 2 (= 2 <sup>1</sup> ) cycles 0x01: 4 (= 2 <sup>2</sup> ) cycles ⋮ 0x1E: 2,147,483,648 (= 2 <sup>31</sup> ) cycles 0x1F: 4,294,967,296 (= 2 <sup>32</sup> ) cycles	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
27:24	PERREP[3:0]	Periodic transaction repeat These bits configure the number of transaction repetitions in periodic manual-command mode. 0x0: 1 (= 2 <sup>0</sup> ) time 0x1: 2 (= 2 <sup>1</sup> ) times ⋮ 0xE: 16384 (= 2 <sup>14</sup> ) times 0xF: 32768 (= 2 <sup>15</sup> ) times	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

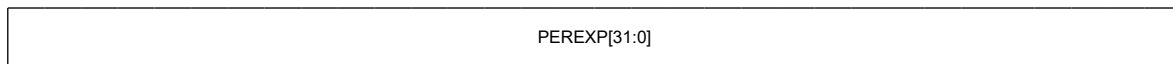
### 35.2.2.5 CDCTL1 : OSPI Command Manual Control Register 1

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x074

Bit position: 31 0

Bit field:



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	PEREXP[31:0]	Periodic transaction expected value These bits configure the expected value to compare with the read value in periodic manual-command mode. For example, in case of comparing 1 byte, the lower byte should be configured.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

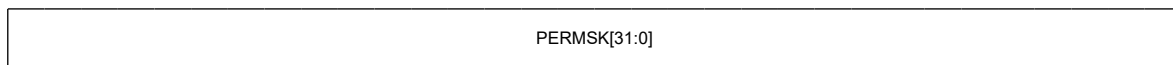
### 35.2.2.6 CDCTL2 : OSPI Command Manual Control Register 2

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x078

Bit position: 31 0

Bit field:



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	PERMSK[31:0]	Periodic transaction masked value These bits configure the masked value for the expected value in periodic manual-command mode. When set 1 to any bit, the corresponding bit configured as expected value (CDCTL1.PEREXP[31:0]) is ignored. In 8D-8D-8D, the data bytes are transferred only in byte pairs on xSPI bus. It means the dummy read data could be stored. It should be masked for unused bits. For example, in case of read lower 1 byte, it should be configured to 0xFFFFFFFF00.	R/W

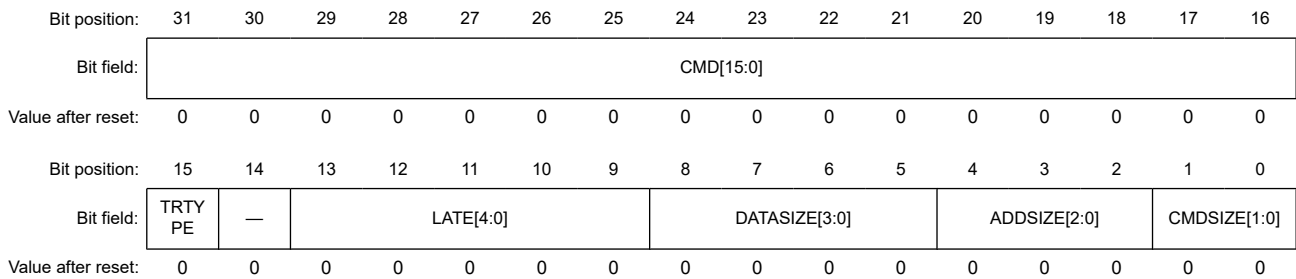
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

### 35.2.2.7 CDTBUFn : OSPI Command Manual Type Buf n (n = 0 to 3)

Base address: OSPI0\_B = 0x4026\_8000  
 OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x080 + 0x010 × n



Bit	Symbol	Function	R/W
1:0	CMDSIZE[1:0]	Command Size These bits configure the size of command field. In case of 8D-8D-8D, it should be fixed to 10b. It should not be configured both command size and address size to zero. 0 0: 0 bytes (No command phase) 0 1: 1 byte 1 0: 2 bytes Others: Setting prohibited	R/W
4:2	ADDSSIZE[2:0]	Address size These bits configure the size of address field. 0 0 0: 0 bytes (No address phase) 0 0 1: 1 byte 0 1 0: 2 bytes 0 1 1: 3 bytes 1 0 0: 4 bytes Others: Setting prohibited	R/W
8:5	DATASIZE[3:0]	Write/Read Data Size These bits configure the size of data field. In 8D-8D-8D, the data bytes are transferred only in byte pairs on xSPI bus. For example, even if configuring 1 byte for read, 2 bytes data is received. The last byte should be ignored. The 0 bytes must not configured for read transaction. 0x0: 0 bytes (No data phase) 0x1: 1 byte ⋮ 0x7: 7 bytes 0x8: 8 bytes Others: Setting prohibited	R/W



Bit	Symbol	Function	R/W
13:9	LATE[4:0]	Latency cycle These bits configure the latency cycle in manual-command mode. 0x0: No latency 0x1: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	TRTYPE	Transaction Type This bit selects the type of transaction. 0: Read transaction (Readout data from slave device) 1: Not read transaction	R/W
31:16	CMD[15:0]	Command (1-2 bytes) These bits configure the command field in manual-command mode. The number of bytes configured in Command Size bit is transferred. 1S-1S-1S, 4S-4D-4D: CMD[15:8] is command field, CMD[7:0] is not used. 8D-8D-8D profile 1.0: CMD[15:8] is command field, CMD[7:0] is extension field. 8D-8D-8D profile 2.0: CMD[15:0] is upper 2 bytes of command & modifier field. (bit 47-32 in xSPI protocol)	R/W

Note: S-TYPE-3, P-TYPE-3

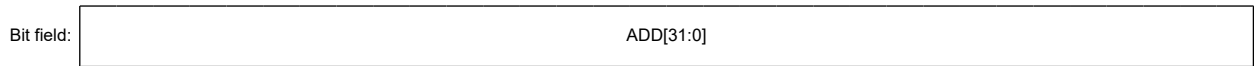
This register has functions to control xSPI Master function.

### 35.2.2.8 CDABUF<sub>n</sub> : OSPI Command Manual Address Buf <sub>n</sub> (n = 0 to 3)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x084 + 0x10 × n

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	ADD[31:0]	Address These bits configure the address field in manual-command mode. 1S-1S-1S, 4S-4D-4D, 8D-8D-8D profile 1.0: It is address field. 8D-8D-8D profile 2.0: It is lower 4 bytes of command & modifier field. (bit 31-0 in xSPI protocol)	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

### 35.2.2.9 CDD0BUF<sub>n</sub> : OSPI Command Manual Data 0 Buf <sub>n</sub> (n = 0 to 3)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x088 + 0x10 × n

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	DATA[31:0]	Write/Read Data These bits configure the data field in manual-command mode. In case of write transaction, the write data should be configured. In case of read transaction, the read data is stored.	R/W

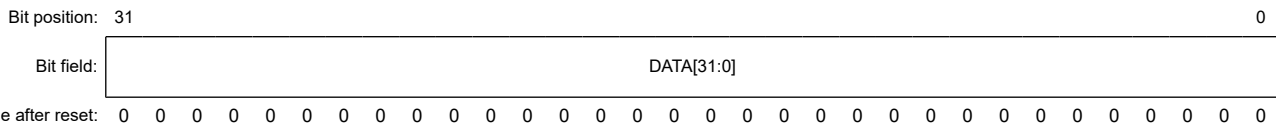
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

### 35.2.2.10 CDD1BUF<sub>n</sub> : OSPI Command Manual Data 1 Buf n (n = 0 to 3)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x08C + 0x10 × n



Bit	Symbol	Function	R/W
31:0	DATA[31:0]	Write/Read Data These bits configure the data field in manual-command mode. In case of write transaction, the write data should be configured. In case of read transaction, the read data is stored.	R/W

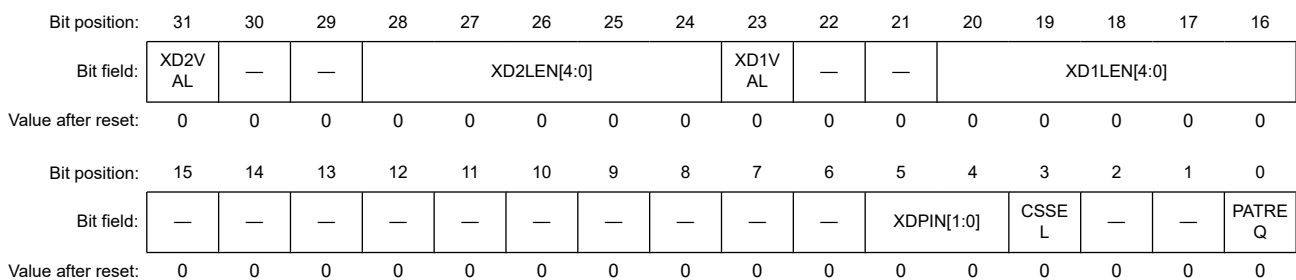
Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

### 35.2.2.11 LPCTL0 : OSPI Link Pattern Control Register 0

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x100



Bit	Symbol	Function	R/W
0	PATREQ	Pattern request This bit requests to issue the pattern. When set to 1, it starts the pattern. It is cleared to 0 when the pattern completed. 0: No request XiP Disable pattern 1: Request XiP Disable pattern	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue a pattern. 0: slave0 (CS0) 1: slave1 (CS1)	R/W

Bit	Symbol	Function	R/W
5:4	XDPIN[1:0]	XiP Disable pattern pin These bits select the data output pins to transmit XiP Disable pattern. 0 0: 1 pin 0 1: 2 pins 1 0: 4 pins 1 1: 8 pins	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
20:16	XD1LEN[4:0]	XiP Disable pattern 1st phase length These bits select the length of 1st phase in XiP disable pattern. The pattern with zero-length both 1st phase and 2nd phase should not be configured. 0x0: 0 cycles 0x1: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	XD1VAL	XiP Disable pattern 1st phase value This bit selects the value of 1st phase in XiP disable pattern. 0: Low drive 1: High drive	R/W
28:24	XD2LEN[4:0]	XiP Disable pattern 2nd phase length These bits select the length of 2nd phase in XiP disable pattern. 0x00: 0 cycles 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
30:29	—	These bits are read as 0. The write value should be 0.	R/W
31	XD2VAL	XiP Disable pattern 2nd phase value This bit selects the value of 2nd phase in XiP disable pattern. 0: Low drive 1: High drive	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

### 35.2.2.12 LPCTL1 : OSPI Link Pattern Control Register 1

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	RSTSU[2:0]			—	RSTWID[2:0]			—	—	RSTREP[1:0]		CSSE L	—	PATREQ[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PATREQ[1:0]	Pattern request These bits request to issue the pattern. When set to 01b or 10b, it starts the pattern. It is cleared to 00b when the pattern completed. 0 0: No request 0 1: Request Reset pattern 1 0: Request CS only pattern 1 1: Setting prohibited	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue a pattern. 0: slave0 (CS0) 1: slave1 (CS1)	R/W
5:4	RSTREP[1:0]	Reset pattern repeat These bits select the repeating time to toggle CS from LOW to HIGH. 0 0: 4 times (Specified on Reset Signaling Protocol) 0 1: 5 times 1 0: 6 times 1 1: 7 times	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RSTWID[2:0]	Reset pattern width These bits configure the width of cycle in reset pattern and CS only pattern. It toggles CS with the configured cycle. 0 0 0: 2 (= 2 <sup>1</sup> ) cycles 0 0 1: 4 (= 2 <sup>2</sup> ) cycles ⋮ 1 1 0: 128 (= 2 <sup>7</sup> ) cycles 1 1 1: 256 (= 2 <sup>8</sup> ) cycles	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	RSTSU[2:0]	Reset pattern data output setup time These bits configure the number of setup cycle for data output based on the edge of CS in reset pattern. It needs enough setup time because xSPI slave samples any data at the rising edge of CS. This cycle of setup time should be less than the cycle of reset pattern width (RSTWID[2:0]). 0 0 0: 1 cycle 0 0 1: 2 cycles ⋮ 1 1 0: 7 cycles 1 1 1: 8 cycles	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

### 35.2.2.13 LIOCTL : OSPI Link I/O Control Register

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSTC S0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WPCS 1	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	—	This bit is read as 1. The write value should be 1.	R/W
1	WPCS1	WP drive for slave1 This bit controls the value of OM_WP1 port. It can be useful only for xSPI slave with write protect port. 0: Drive Low level 1: Drive High level	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
16	RSTCS0	Reset drive This bit controls the value of OM_RESET port. It could be useful only for xSPI slave with reset port. 0: Drive Low level 1: Drive High level	R/W
17	—	This bit is read as 1. The write value should be 1.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

### 35.2.2.14 CCCTL0CSn : OSPI Command Calibration Control Register 0 CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x130 + 0x020 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CASFTEND[4:0]				—	—	—	CASFTSTA[4:0]					
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CAITV[4:0]				—	—	—	—	—	—	CANO WR	CAEN	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CAEN	Automatic Calibration Enable This bit enables the automatic calibration. When set to 1, it transmits the calibration sequence periodically and adjusts the value of phase shift. When set to 0 during the calibration sequence, it stops after completed ongoing calibration sequence, and then this bit is cleared. 0: Disable automatic calibration 1: Enable automatic calibration	R/W
1	CANOWR	Calibration no write mode This bit selects to omit write command in calibration sequence. It can be used for any slave device with fixed calibration pattern data. 0: Calibration sequence with write command 1: Calibration sequence without write command	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
12:8	CAITV[4:0]	Calibration interval These bits configure the interval between calibration patterns. 0x00: 2 (= 2 <sup>1</sup> ) cycle wait 0x01: 4 (= 2 <sup>2</sup> ) cycle wait ⋮ 0x1E: 2,147,483,648 (= 2 <sup>31</sup> ) cycle wait 0x1F: 4,294,967,296 (= 2 <sup>32</sup> ) cycle wait	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20:16	CASFTSTA[4:0]	Calibration OM_DQS shift start value These bits configure the start value of OM_DQS shift.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	CASFTEND[4:0]	Calibration OM_DQS shift end value These bits configure the end value of OM_DQS shift. It should be equal or more than the start value (CASFTSTA[4:0]).	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function.

### 35.2.2.15 CCCTL1CSn : OSPI Command Calibration Control Register 1 CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x134 + 0x020 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CARDLATE[4:0]				—	—	—	CAWRLATE[4:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CADATASIZE[3:0]			CAADDSIZE[2:0]			CACMDSIZE[1:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CACMDSIZE[1:0]	Command Size These bits configure the size of command field. In case of 8D-8D-8D, it should be fixed to 10b. It should not be configured both command size and address size to zero. 0 0: 0 bytes (No command phase) 0 1: 1 byte 1 0: 2 bytes 1 1: Setting prohibited	R/W
4:2	CAADDSIZE[2:0]	Address size These bits configure the size of address field. 0 0 0: 0 bytes (No address phase) 0 0 1: 1 byte 0 1 0: 2 bytes 0 1 1: 3 bytes 1 0 0: 4 bytes Others: Setting prohibited	R/W
8:5	CADATASIZE[3:0]	Write/Read Data Size These bits configure the size of data field. In 8D-8D-8D, it should be configured with even byte. 0x0: 1 byte 0x1: 2 bytes ⋮ 0xE: 15 bytes 0xF: 16 bytes	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20:16	CAWRLATE[4:0]	Write Latency cycle These bits configure the latency cycle in calibration frame. 0x00: No latency 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	CARDLATE[4:0]	Read Latency cycle These bits configure the latency cycle in calibration frame. 0x00: No latency 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

### 35.2.2.16 CCCTL2CSn : OSPI Command Calibration Control Register 2 CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x138 + 0x020 × n

Bit position: 31 16 15 0



Value after reset: 0

Bit	Symbol	Function	R/W
15:0	CAWRCMD[15:0]	Calibration pattern write command These bits configure the calibration pattern write command.	R/W
31:16	CARDCMD[15:0]	Calibration pattern read command These bits configure the calibration pattern read command.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

### 35.2.2.17 CCCTL3CSn : OSPI Command Calibration Control Register 3 CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x13C + 0x020 × n

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CAADD[31:0]	Calibration pattern address These bits configure the calibration pattern address.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

### 35.2.2.18 CCCTL4CSn : OSPI Command Calibration Control Register 4 CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x140 + 0x020 × n

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

Note: S-TYPE-3, P-TYPE-3

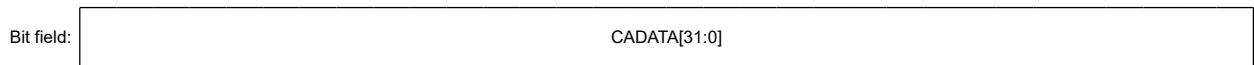
This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

### 35.2.2.19 CCCTL5CSn : OSPI Command Calibration Control Register 5 CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x144 + 0x020 × n

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

Note: S-TYPE-3, P-TYPE-3

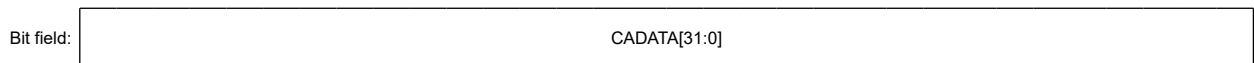
This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

### 35.2.2.20 CCCTL6CSn : OSPI Command Calibration Control Register 6 CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x148 + 0x020 × n

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.



### 35.2.2.21 CCCTL7CSn : OSPI Command Calibration Control Register 7 CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x14C + 0x020 × n

Bit position: 31

0

Bit field:

CADATA[31:0]

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

Note: S-TYPE-3, P-TYPE-3

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

### 35.2.3 OSPI Status Registers

These registers monitor the status of xSPI Master.

#### 35.2.3.1 COMSTT : OSPI Common Status Register

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x184

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	RSTO CS1	INTCS 1	ECSC S1	—	—	—	—

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	WRBU FNEC H0	—	PBUF NECH 0	—	—	—	MEMA CCCH 0

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	MEMACCCH0	Memory access ongoing from ch0 This bit is only valid in the <a href="#">section 35.3.7.2. Flow of Communication Stop</a> and <a href="#">section 35.3.7.6. Flow of Memory-mapping Stop</a> flows. 0: System bus bridge ch0 is not accessing to memory. 1: System bus bridge ch0 is accessing to memory.	R
3:1	—	These bits are read as 0.	R
4	PBUFNECH0	Prefetch Buffer Not Empty for ch0 0: Empty 1: Not empty	R
5	—	This bit is read as 0.	R
6	WRBUFNECH0	Write Buffer Not Empty for ch0 This bit is only valid in the <a href="#">section 35.3.7.2. Flow of Communication Stop</a> and <a href="#">section 35.3.7.6. Flow of Memory-mapping Stop</a> flows. 0: Empty 1: Not empty	R
19:7	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
20	ECSCS1	ECS monitor for slave1 This bit indicates the value of OM_ECSINT1 port. 0: Low level 1: High level	R
21	INTCS1	INT monitor for slave1 This bit indicates the value of OM_ECSINT1 port. 0: Low level 1: High level	R
22	RSTOCS1	RSTO monitor for slave1 This bit indicates the value of OM_RSTO1 port. 0: Low level 1: High level	R
31:23	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

This register indicates the status of xSPI Master.

### 35.2.3.2 CASTTCSn : OSPI Calibration Status Register CSn (n = 0, 1)

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x188 + 0x004 × n

Bit position: 31

0

Bit field:

CASUC[31:0]

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CASUC[31:0]	Calibration Success These bits indicate the calibration success for each OM_DQS shift value. It is updated when completed each calibration sequence. CASUC[x] indicates calibration success in OM_DQS shift value = x.	R

Note: S-TYPE-3, P-TYPE-3

This register indicates the status of xSPI Master.

## 35.2.4 OSPI Interrupt Registers

These registers control the interrupt function of xSPI Master.

### 35.2.4.1 INTS : OSPI Interrupt Status Register

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x190

Bit position: 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16

Bit field:

CASU CCS1	CASU CCS0	CAFAI LCS1	CAFAI LCS0	—	—	—	—	—	—	—	BUSE RRCH 0	—	—	—	—
--------------	--------------	---------------	---------------	---	---	---	---	---	---	---	-------------------	---	---	---	---

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:

—	—	INTCS 1	—	—	—	ECSC S1	—	—	—	DSTO CS1	DSTO CS0	PERT O	—	PATC MP	CMDC MP
---	---	------------	---	---	---	------------	---	---	---	-------------	-------------	-----------	---	------------	------------

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	CMDCMP	Command Completed This bit is set to 1 when completed the requested manual-command. In direct manual-command, it means all transactions completed. In periodic manual-command, it means the read data matches with the expected data. 0: No detection 1: Detection	R
1	PATCMP	Pattern Completed This bit is set to 1 when completed the requested pattern. 0: No detection 1: Detection	R
2	—	This bit is read as 0.	R
3	PERTO	Periodic transaction timeout This bit is set to 1 when the read value does not match with the expected value in periodic manual-command mode. 0: No detection 1: Detection	R
4	DSTOCS0	OM_DQS timeout for slave0 This bit is set to 1 when lost OM_DQS in read transaction with using OM_DQS. It means not receiving the data during expected read phase. In this case, xSPI master stops the read transaction and the following transaction. This error may issue error response to system bus. 0: No detection 1: Detection	R
5	DSTOCS1	OM_DQS timeout for slave1 This function is same as DSTOCS0. 0: No detection 1: Detection	R
8:6	—	These bits are read as 0.	R
9	ECSCS1	ECC error detection for slave1 This bit is set to 1 when detected the falling edge on OM_ECSINT1 port. It can be useful only for xSPI slave with ECC detection function. 0: No detection 1: Detection	R
12:10	—	These bits are read as 0.	R
13	INTCS1	Interrupt detection for slave1 This bit is set to 1 when detected the falling edge on OM_ECSINT1 port. It can be useful only for xSPI slave with interrupt function. 0: No detection 1: Detection	R
19:14	—	These bits are read as 0.	R
20	BUSERRCH0	System bus error for ch0 This bit is set to 1 when an error response occurs on system bus channel0.	R
27:21	—	These bits are read as 0.	R
28	CAFAILCS0	Calibration failed for slave0 This bit is set to 1 when failed calibration. 0: No detection 1: Detection	R
29	CAFAILCS1	Calibration failed for slave1 This function is same as CAFAILCS0. 0: No detection 1: Detection	R
30	CASUCCS0	Calibration success for slave0 This bit is set to 1 when success calibration. 0: No detection 1: Detection	R

Bit	Symbol	Function	R/W
31	CASUCCS1	Calibration success for slave1 This function is same as CASUCCS0. 0: No detection 1: Detection	R

Note: S-TYPE-3, P-TYPE-3

This register indicates the status of interrupt. The bits in this register are cleared to 0 when writing 1 on the corresponding bit of INTC register.

### 35.2.4.2 INTC : OSPI Interrupt Clear Register

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x194

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASU CCS1 C	CASU CCS0 C	CAFAI LCS1 C	CAFAI LCS0 C	—	—	—	—	—	—	—	BUSE RRCH 0C	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS 1C	—	—	—	ECSC S1C	—	—	—	DSTO CS1C	DSTO CS0C	PERT OC	—	PATC MPC	CMDC MPC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMPC	Command Completed interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
1	PATCMPC	Pattern Completed interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
2	—	The write value should be 0.	W
3	PERTOC	Periodic transaction timeout interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
4	DSTOCS0C	OM_DQS timeout for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
5	DSTOCS1C	OM_DQS timeout for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
8:6	—	The write value should be 0.	W
9	ECSCS1C	ECC error detection for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
12:10	—	The write value should be 0.	W
13	INTCS1C	Interrupt detection for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
19:14	—	The write value should be 0.	W
20	BUSERRCH0C	System bus error for ch0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
27:21	—	The write value should be 0.	W

Bit	Symbol	Function	R/W
28	CAFAILCS0C	Calibration failed for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
29	CAFAILCS1C	Calibration failed for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
30	CASUCCS0C	Calibration success for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
31	CASUCCS1C	Calibration success for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W

Note: S-TYPE-3, P-TYPE-3

This register clears the status of interrupt.

### 35.2.4.3 INTE : OSPI Interrupt Enable Register

Base address: OSPI0\_B = 0x4026\_8000  
OSPI0\_B\_NS = 0x5026\_8000

Offset address: 0x198

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASU CCS1 E	CASU CCS0 E	CAFAI LCS1E	CAFAI LCS0E	—	—	—	—	—	—	—	BUSE RRCH OE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS 1E	—	—	—	ECSC S1E	—	—	—	DSTO CS1E	DSTO CS0E	PERT OE	—	PATC MPE	CMDC MPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMPE	Command Completed interrupt enable 0: Disabled 1: Enabled	R/W
1	PATCMPE	Pattern Completed interrupt enable 0: Disabled 1: Enabled	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	PERTOE	Periodic transaction timeout interrupt enable 0: Disabled 1: Enabled	R/W
4	DSTOCS0E	OM_DQS timeout for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
5	DSTOCS1E	OM_DQS timeout for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
8:6	—	These bits are read as 0. The write value should be 0.	R/W
9	ECSCS1E	ECC error detection for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
13	INTCS1E	Interrupt detection for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
19:14	—	These bits are read as 0. The write value should be 0.	R/W
20	BUSERRCH0E	System bus error for ch0 interrupt enable 0: Disabled 1: Enabled	R/W
27:21	—	These bits are read as 0. The write value should be 0.	R/W
28	CAFAILCS0E	Calibration failed for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
29	CAFAILCS1E	Calibration failed for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
30	CASUCCS0E	Calibration success for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
31	CASUCCS1E	Calibration success for slave1 interrupt enable 0: Disabled 1: Enabled	R/W

Note: S-TYPE-3, P-TYPE-3

This register enables the interrupt.

### 35.3 Operation

xSPI Master interface has the functions to issue the transaction for external memory with xSPI Slave interface. It allows to write to registers in external memory or read from it.

This xSPI Master mainly has two modes to issue the transaction. One is a manual-command mode; Software configures all fields of xSPI frame and starts the transaction by software request. The other is a memory-mapping mode; it automatically converts system bus for pre-configured memory area into xSPI transaction. It enables to access from system bus to external memory area outside of chip via xSPI bus.

This section describes the xSPI bus operation, the direct control of xSPI frame (manual-command), the control of memory access (memory-mapping), the error operation, and the flow to operation.

#### 35.3.1 xSPI Bus

This section describes the xSPI bus operation.

Figure 35.2 shows an example of connections between OSPI and memory devices.

In this case, OM\_CS0 should be connected to the RAM device and OM\_CS1 should be connected to the Flash device.

Pull-up resistors should be specified according to the instructions for each device.

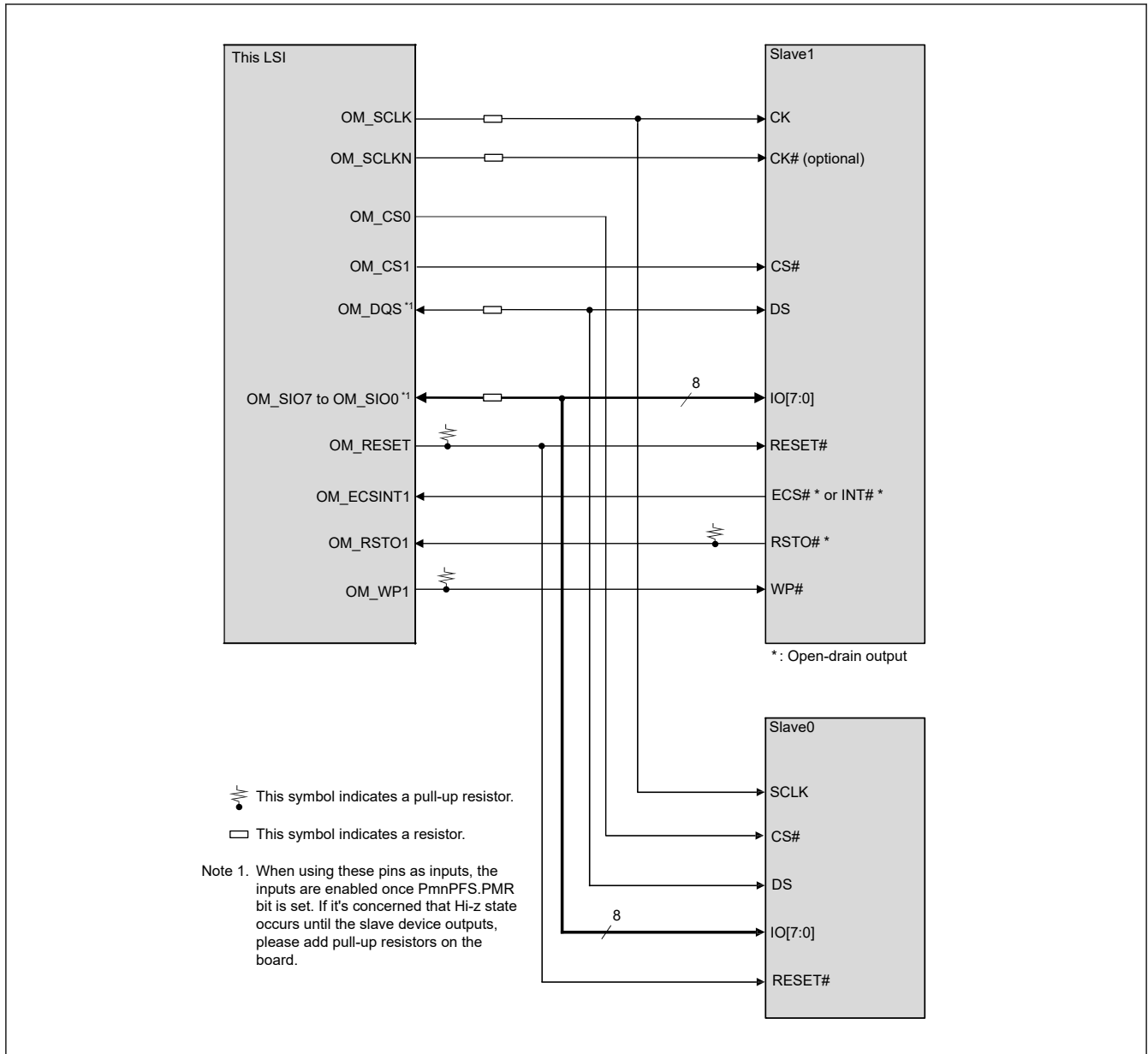


Figure 35.2 Example of Connection between this and memory devices

### 35.3.1.1 Supported Protocol Mode

This xSPI Master supports various protocol modes. It is configured by Protocol mode bits (LIOCFCGSn.PRTMD[9:0]). Table 35.3 shows the summary of protocol modes.

Table 35.3 Supported protocol mode (1 of 2)

Protocol mode	Function	PRTMD[9:0]	Note
1S-1S-1S	Command, Address and Data field are transferred at SDR with using 1 data input pin and 1 data output pin. Read data is sampled with CK.	0x000	Specified by xSPI protocol
4S-4D-4D	Command field is transferred at SDR with using 4 data pins. Address and Data fields are transferred at DDR with using 4 data pins. Read data is sampled with OM_DQS.	0x3B2	Specified by xSPI protocol
8D-8D-8D	Command, Address and Data fields are transferred at DDR with using 8 data pins. Read data is sampled with OM_DQS.	0x3FF	Specified by xSPI protocol
1S-2S-2S	Command field is transferred at SDR with using 1 data pin. Address and Data fields are transferred at SDR with using 2 data pins. Read data is sampled with CK.	0x048	—

**Table 35.3 Supported protocol mode (2 of 2)**

Protocol mode	Function	PRTMD[9:0]	Note
2S-2S-2S	Command, Address and Data fields are transferred at SDR with using 2 data pins. Read data is sampled with CK.	0x049	—
1S-4S-4S	Command field is transferred at SDR with using 1 data pin. Address and Data fields are transferred at SDR with using 4 data pins. Read data is sampled with CK.	0x090	—
4S-4S-4S	Command, Address, and Data fields are transferred at SDR with using 4 data pins. Read data is sampled with CK.	0x092	—

Note: In case of XiP mode enable, XiP code is inserted in Latency field. It is valid only for memory-mapping mode.

The bytes of Command and Address fields are transferred in highest order to lowest order sequence. The sequential bytes of Data field are transferred in lowest address to highest address order. In case of using multiple pins, the least significant bit of each byte is placed on OM\_SIO0 with each higher order bit on the successively higher numbered OM\_SIO<sub>n</sub> (n = 1 to 7) signals.



Figure 35.3 shows timing-chart for 1S-1S-1S protocol mode. The OM\_SIO0 signal is used for output data and the OM\_SIO1 signal is used for input data.

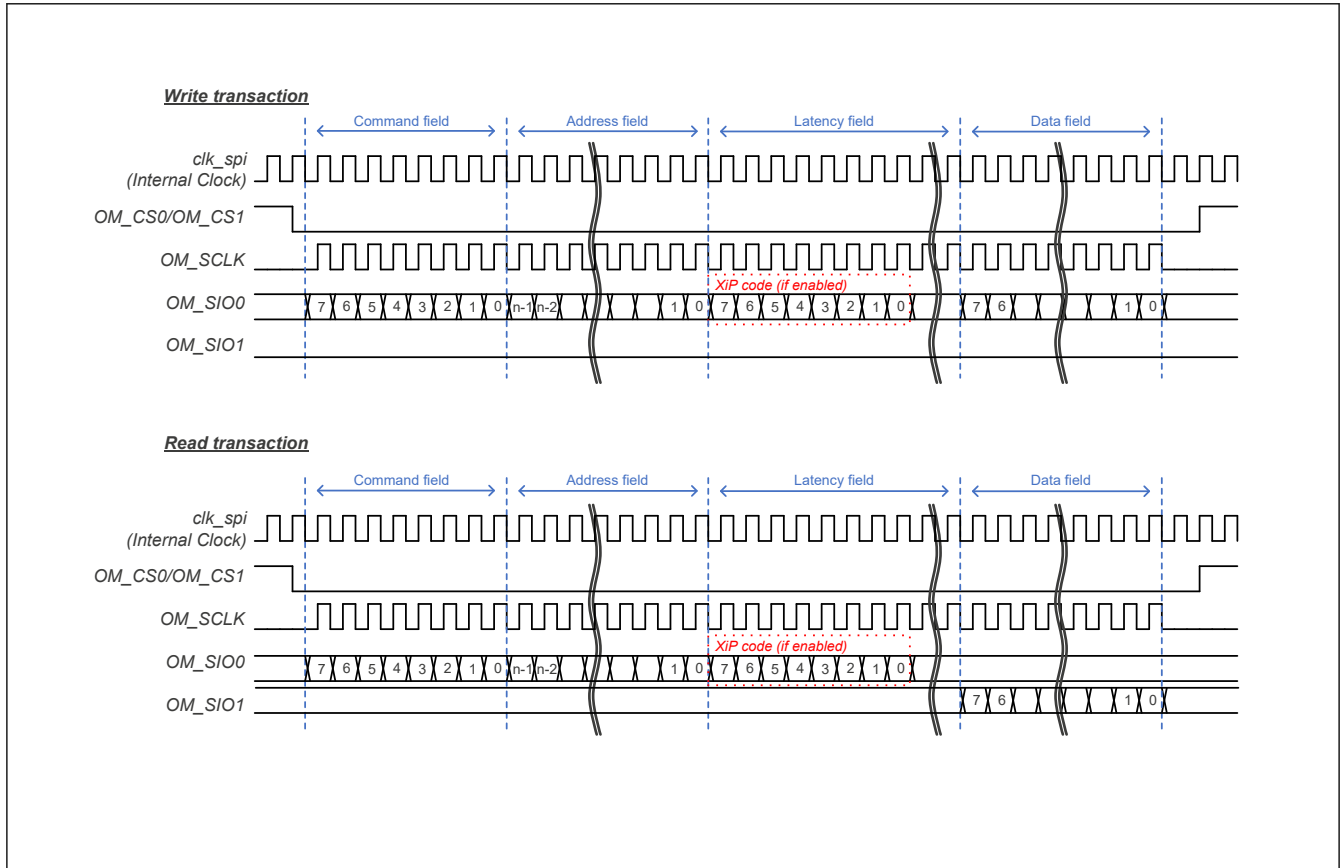


Figure 35.3 1S-1S-1S timing-chart

Figure 35.4 shows timing-chart for 1S-2S-2S protocol mode.



Figure 35.4 1S-2S-2S timing-chart

Figure 35.5 shows timing-chart for 4S-4D-4D protocol mode.

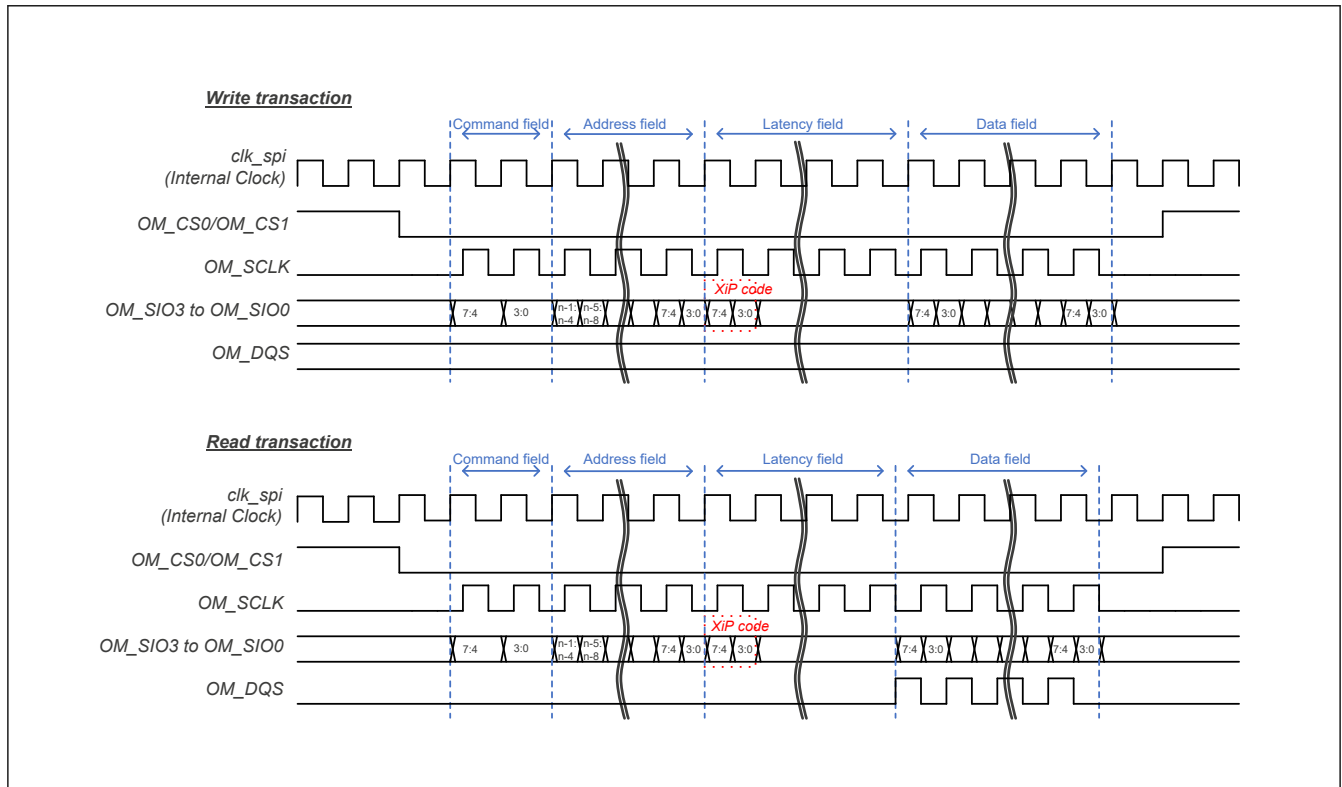


Figure 35.5 4S-4D-4D timing-chart

Figure 35.6 shows timing-chart for 8D-8D-8D profile 1.0 protocol mode. In 8D-8D-8D, the data is always transmitted with byte-pair. In case each field is odd byte, last one byte is padded with invalid data.

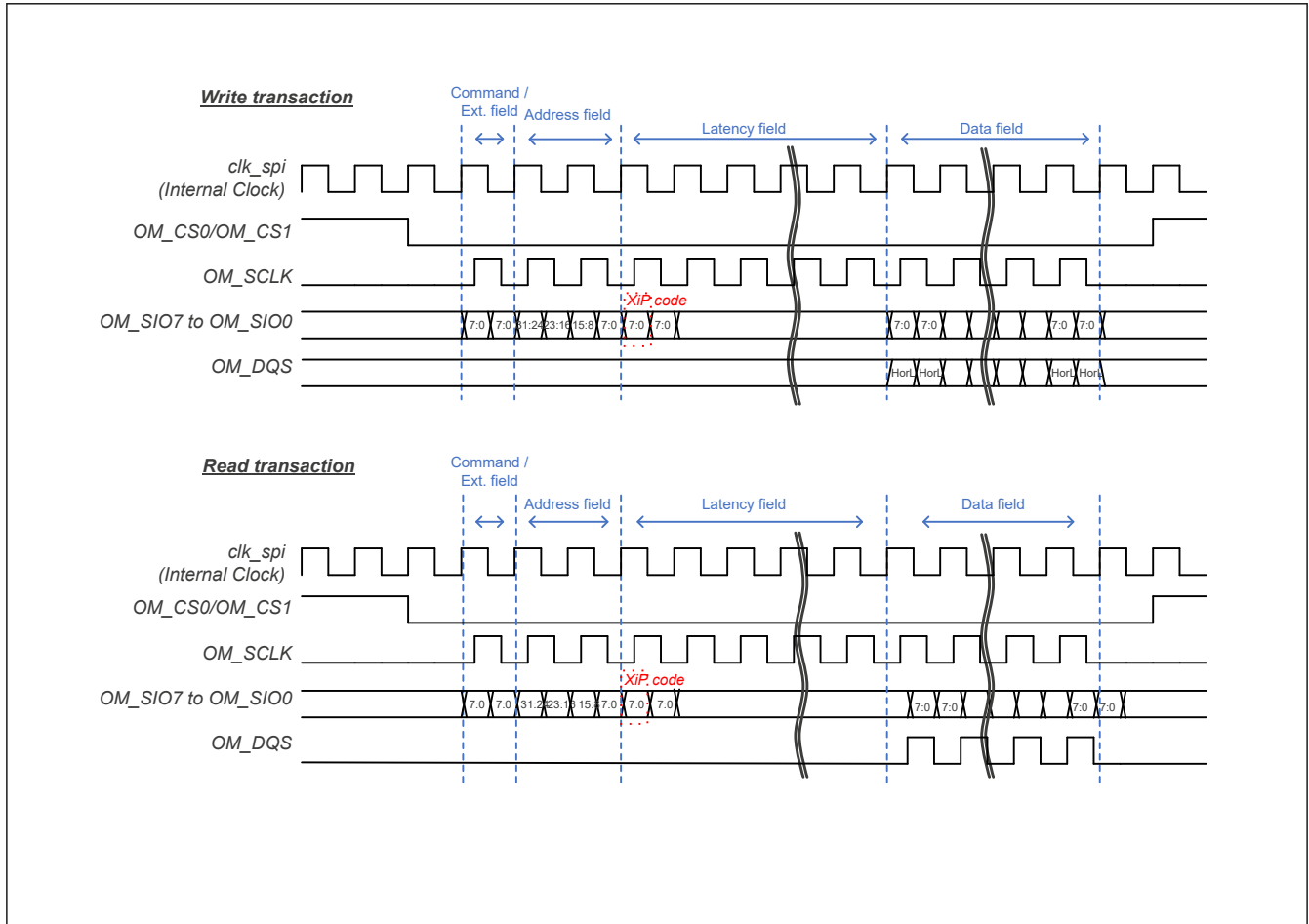


Figure 35.6 8D-8D-8D profile 1.0 timing-chart

Figure 35.7 shows timing-chart for 8D-8D-8D profile 2.0 protocol mode. In 8D-8D-8D, the data is always transmitted with byte-pair. In case each field is odd byte, last one byte is padded with invalid data.

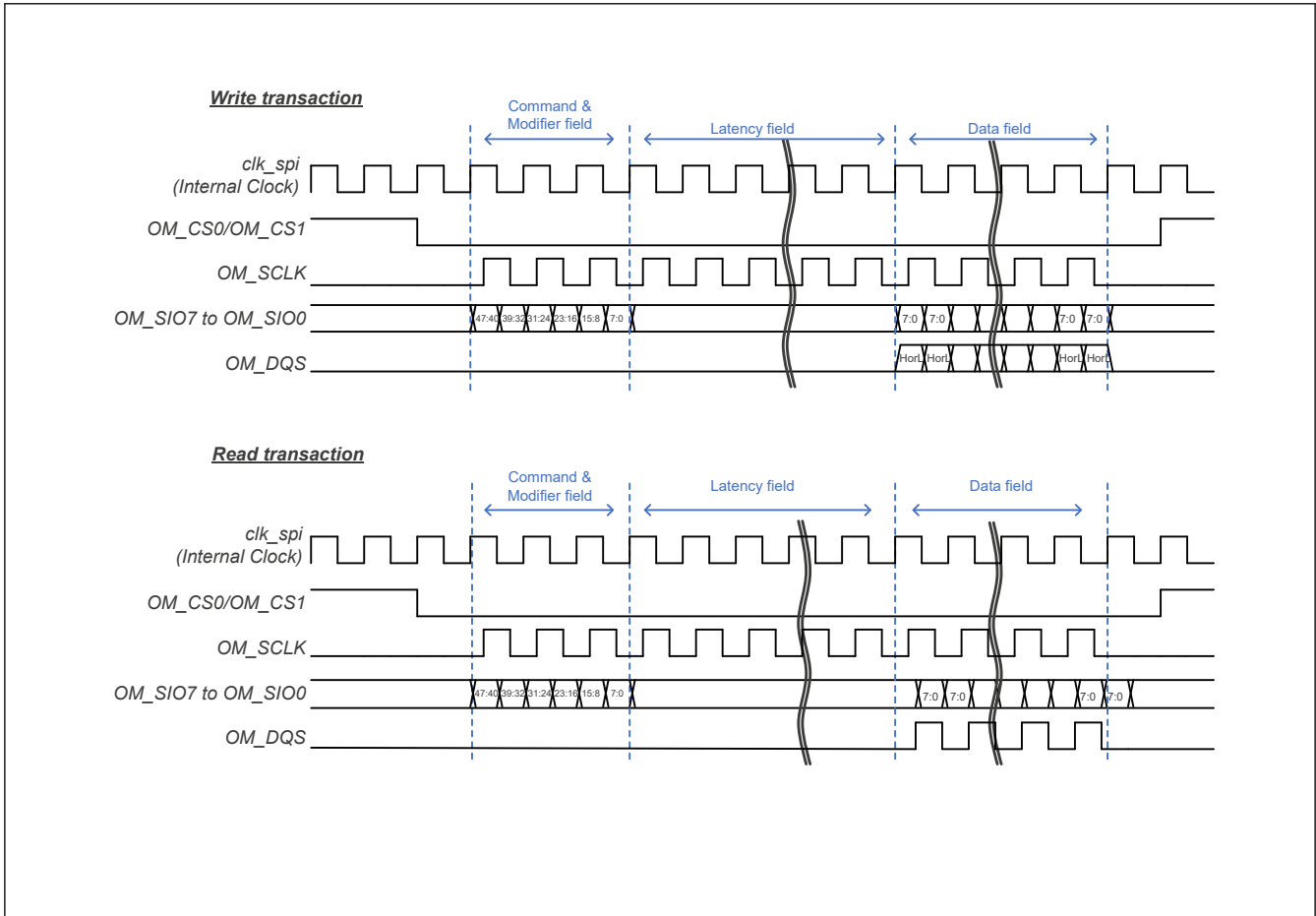


Figure 35.7 8D-8D-8D profile 2.0 timing-chart

### 35.3.1.2 xSPI Frame Interval

The interval between xSPI frames is configured with CS minimum idle term bits (LIOCFGCSn.CSMIN[3:0]). It depends on the specification of xSPI slave device.

### 35.3.1.3 OSPI Signals Timing Control

This xSPI Master supports both SDR and DDR. It is possible to sample input data with Data-Strobe (OM\_DQS) signal at SDR. For various modes and easy implementation, this xSPI Master could adjust the timing to drive/sample xSPI interface signals statically. Table 35.4 shows the summary of xSPI Interface signal timing control.

Table 35.4 Summary of OSPI signals timing control (1 of 2)

Signal	Mode	Default operation	Timing control (n = 0, 1)
OM_CS0, OM_CS1 drive	—	Asserting 1 cycle before the rising-edge of first OM_SCLK	1 cycle extension for asserting with LIOCFGCSn.CSASTEX bit
		Negating 1.5 cycle after the falling-edge of last OM_SCLK	1 cycle extension for negating with LIOCFGCSn.CSNEGEX bit
OM_SCLK drive	SDR without OM_DQS	Reference point	—
	SDR with OM_DQS		
	DDR with OM_DQS	Reference point	—

**Table 35.4 Summary of OSPI signals timing control (2 of 2)**

Signal	Mode	Default operation	Timing control (n = 0, 1)
OM_SIO <sub>n</sub> (n = 0 to 7) output drive	SDR without OM_DQS	Falling edge of clk_spi (Internal Clock)	0 or 0.5 cycle shift with LIOCFGCSn.SDRDRV bit
	SDR with OM_DQS		
	DDR with OM_DQS	Both edges of clk_spi (Internal Clock)	—
OM_SIO <sub>n</sub> (n = 0 to 7) input sample	SDR without OM_DQS	Falling edge of OM_SCLK on expected data size	0 to 7 cycle shift (1 cycle unit) with LIOCFGCSn.SDRSMPSFT[3:0] bits 0 or 0.5 cycle shift with LIOCFGCSn.SDRSMPMD bit
	SDR with OM_DQS	Falling edge of OM_DQS signal on expected data size	Sample at rising edge with LIOCFGCSn.SDRSMPMD bit 0 to 1 cycle phase shift with WRAPCFG.DSSFTCSn[4:0] bits 0 to 7 cycle extension with LIOCFGCSn.DDRSMPEX[3:0] bits
	DDR with OM_DQS	Both edges of OM_DQS signal on expected data size	0 to 1 cycle phase shift with WRAPCFG.DSSFTCSn[4:0] bits 0 to 7 cycle extension with LIOCFGCSn.DDRSMPEX[3:0] bits

Note: In DDR on xSPI protocol, OM\_DQS should be aligned for center of data. It means to shift the phase by 0.25 cycle (90 degrees). This xSPI master supports to adjust this phase depending on the usage conditions.

Figure 35.8 shows the default operation and timing control for SDR without OM\_DQS.

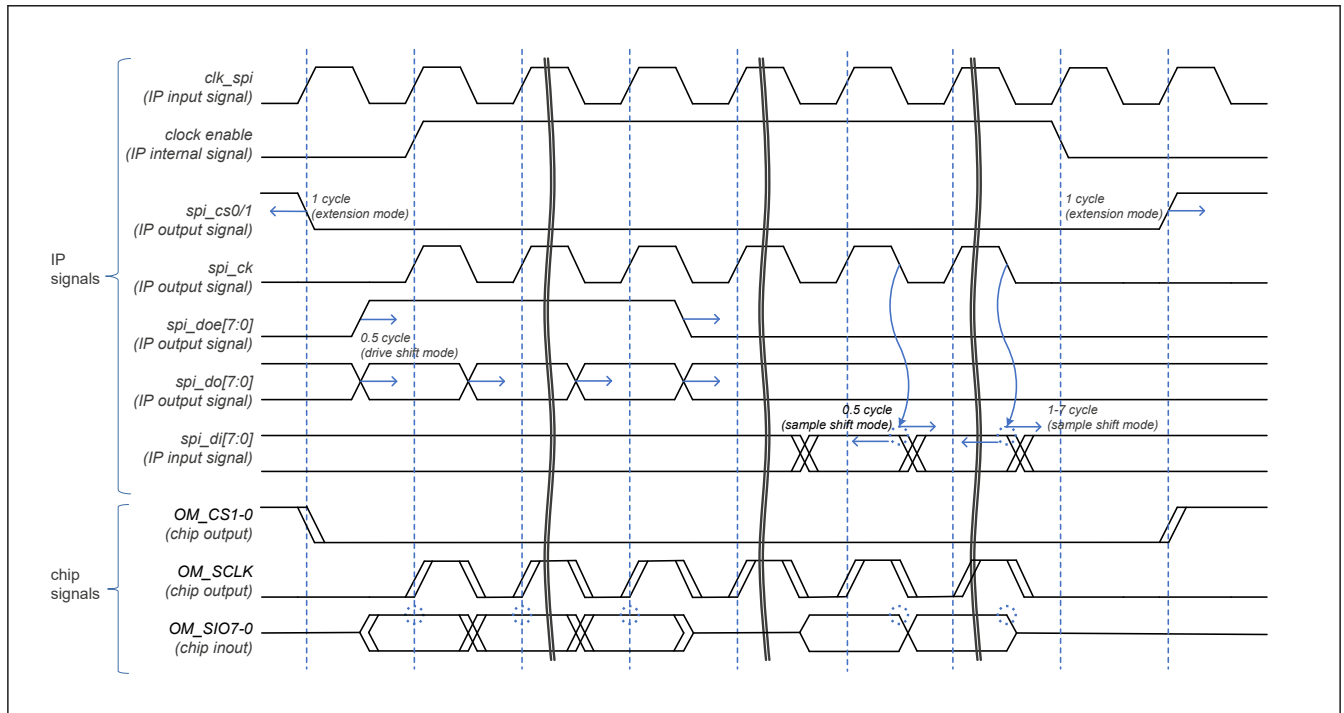


Figure 35.8 Timing control for SDR without OM\_DQS

Figure 35.9 shows the default operation and timing control for SDR with OM\_DQS.

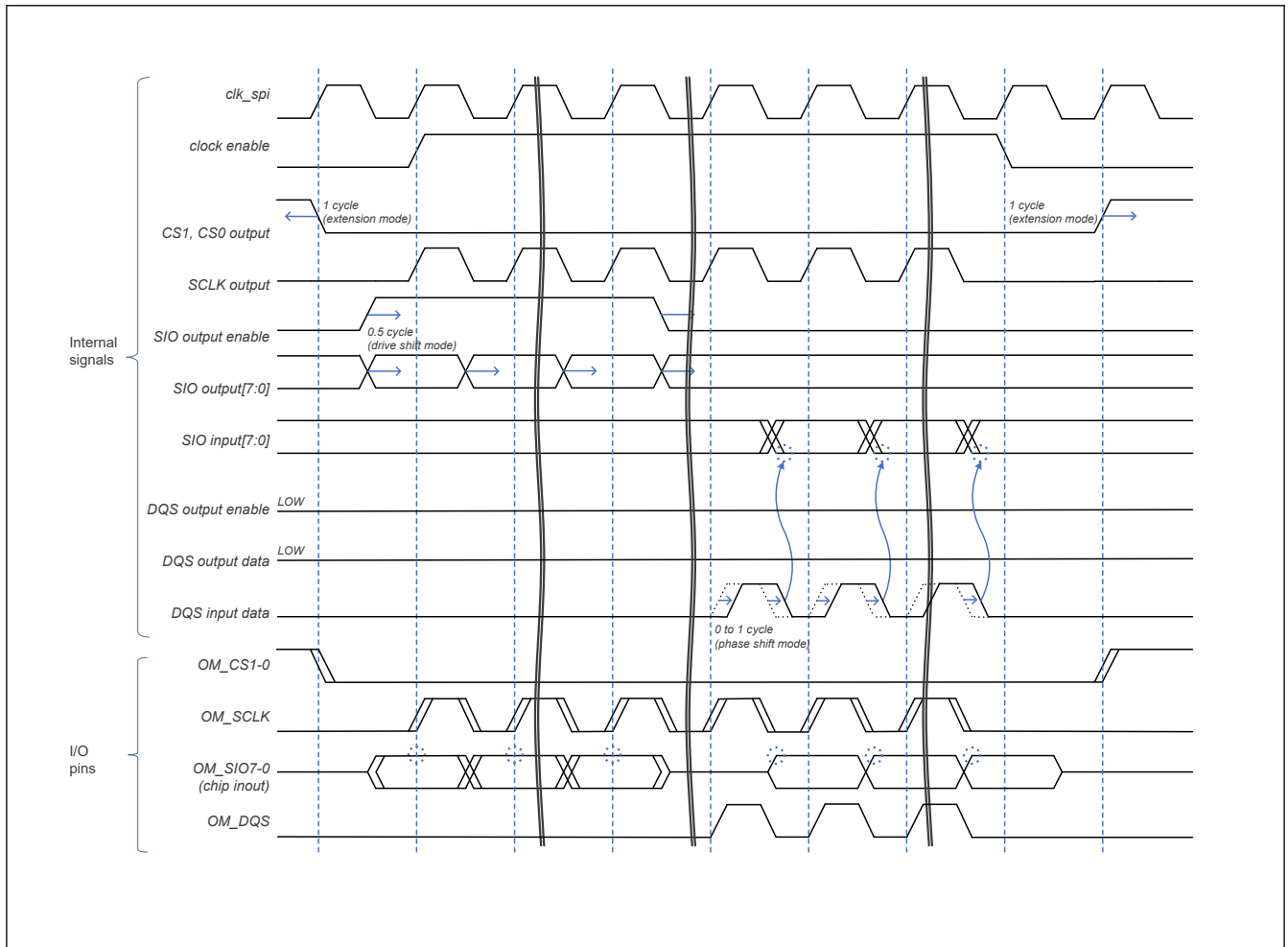


Figure 35.9 Timing control for SDR with OM\_DQS



Figure 35.10 shows the default operation and timing control for DDR with OM\_DQS.

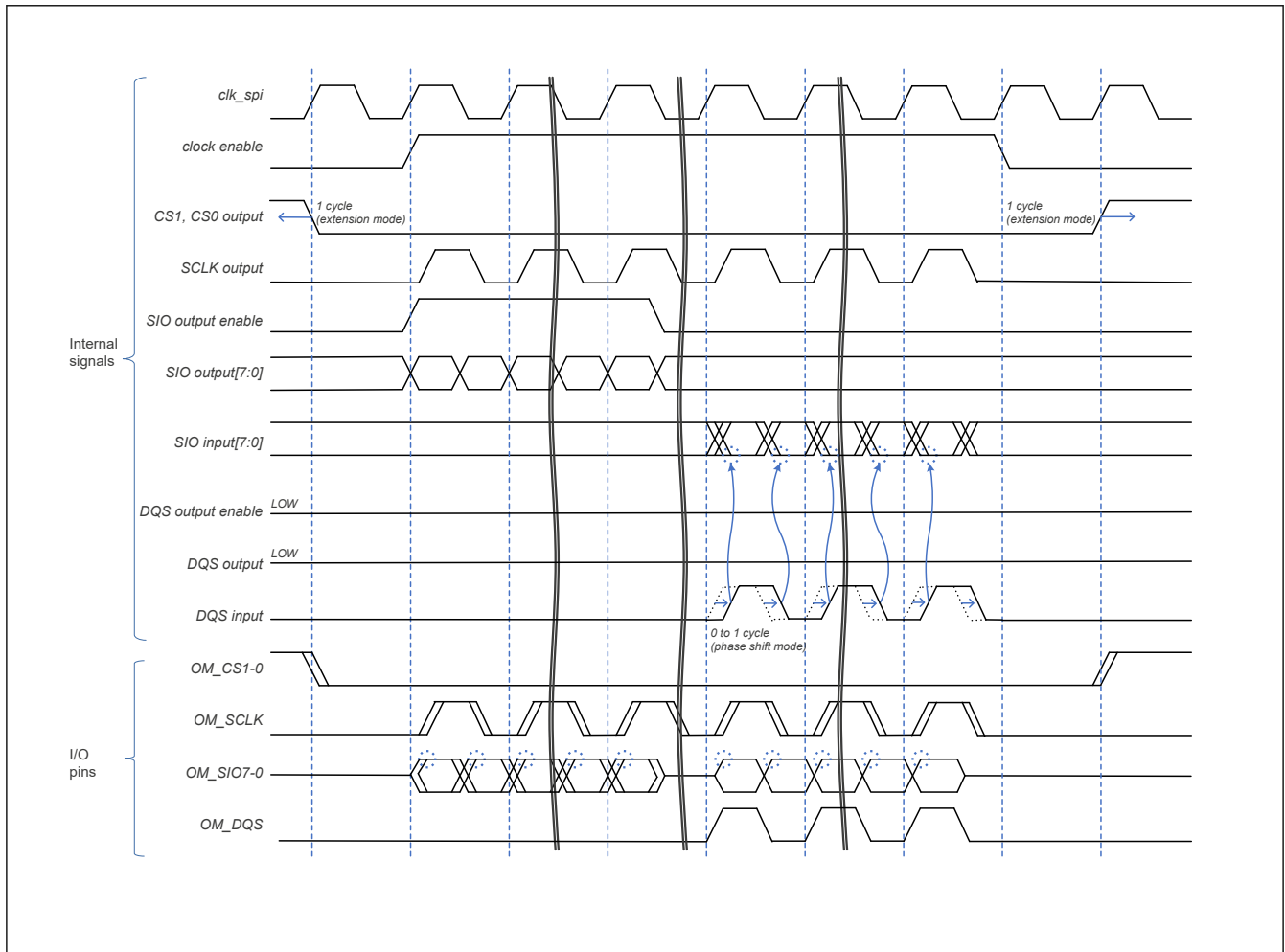


Figure 35.10 Timing control for DDR with OM\_DQS

### 35.3.1.4 Automatic calibration

This xSPI master supports the function to adjust OM\_DQS shift value (WRAPCFG.DSSFTCSn (n = 0, 1)) automatically. When this function is enabled (CCCTL0CSn.CAEN = 1), this xSPI master transmits the calibration sequence periodically and adjusts the value of phase shift.

When all read compare is mismatched in read transaction during the calibration sequence, the calibration fail bit (INTS.CAFAILCSn = 1 (n = 0, 1)) is asserted and OM\_DQS shift value is not updated. When at least one read compare is matched, the calibration success bit (INTS.CASUCCESSn = 1 (n = 0, 1)) is asserted and OM\_DQS shift value is updated. The result of each OM\_DQS shift value is monitored by Calibration Status register (CASTTCSn). Figure 35.11 shows automatic calibration.

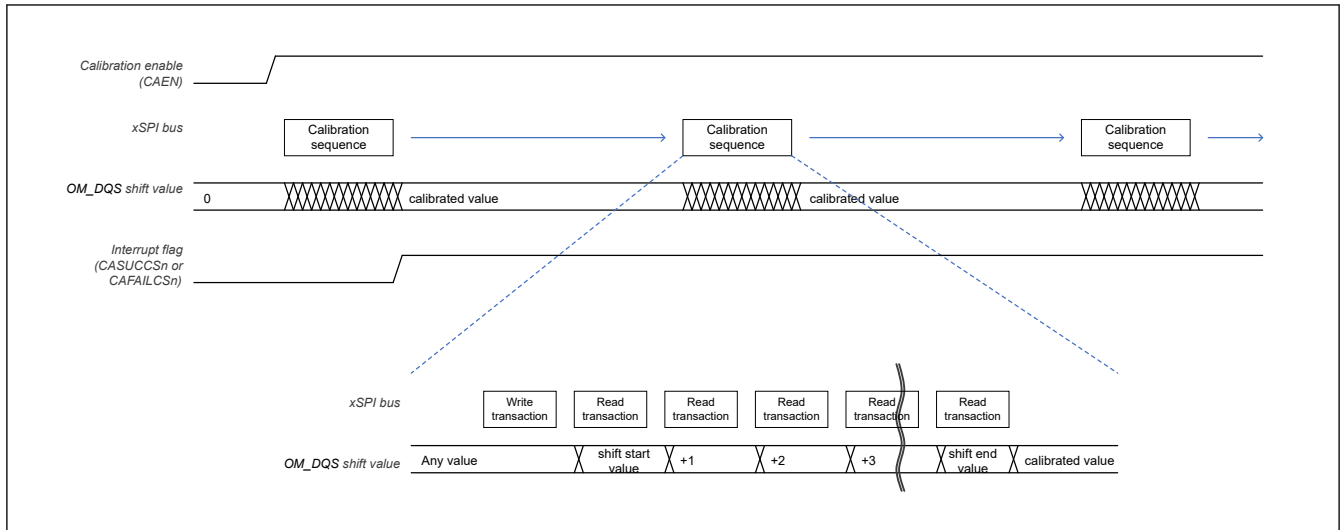


Figure 35.11 Automatic calibration

### 35.3.2 Manual-command

This section describes the manual-command mode. The manual-command has two functional modes: direct mode and periodic mode.

#### 35.3.2.1 Direct Mode

This mode sequentially can issue up to four xSPI transactions configured and requested by Software. A series of transaction can be issued by a transaction request (CDCTL0.TRREQ = 1 with PERMD = 0). The number of transactions (CDCTL0.TRNUM[1:0]) can be configured up to 4. It can be used to change the mode or read the status of xSPI slave device.

Table 35.5 shows the configured register bits for direct manual-command. The operating flow is illustrated in Figure 35.25.

Table 35.5 Manual-command configuration for direct mode

Transaction	Transaction type	Command	Command size	Address	Address size	Data (x = 0, 1)	Data size	Latency cycle
1st Transaction	CDTBUF0. TRTYPE	CDTBUF0. CMD[15:0]	CDTBUF0. CMDSIZE[1:0]	CDABUF0. ADD[31:0]	CDTBUF0. ADDSIZE[2:0]	CDDxBUF0. DATA[31:0]	CDTBUF0. DATASIZE[3:0]	CDTBUF0. LATE[4:0]
2nd Transaction	CDTBUF1. TRTYPE	CDTBUF1. CMD[15:0]	CDTBUF1. CMDSIZE[1:0]	CDABUF1. ADD[31:0]	CDTBUF1. ADDSIZE[2:0]	CDDxBUF1. DATA[31:0]	CDTBUF1. DATASIZE[3:0]	CDTBUF1. LATE[4:0]
3rd Transaction	CDTBUF2. TRTYPE	CDTBUF2. CMD[15:0]	CDTBUF2. CMDSIZE[1:0]	CDABUF2. ADD[31:0]	CDTBUF2. ADDSIZE[2:0]	CDDxBUF2. DATA[31:0]	CDTBUF2. DATASIZE[3:0]	CDTBUF2. LATE[4:0]
4th Transaction	CDTBUF3. TRTYPE	CDTBUF3. CMD[15:0]	CDTBUF3. CMDSIZE[1:0]	CDABUF3. ADD[31:0]	CDTBUF3. ADDSIZE[2:0]	CDDxBUF3. DATA[31:0]	CDTBUF3. DATASIZE[3:0]	CDTBUF3. LATE[4:0]

#### 35.3.2.2 Periodic Mode

This mode periodically issues an xSPI read transaction configured and requested by Software. And it can compare the read value up to 4 bytes with expected value. The transaction is issued by a transaction request (CDCTL0.TRREQ = 1 with PERMD = 1). It can be used to alternate the status polling operation of xSPI slave device.

The periodic term is configured in Periodic transaction interval bits (CDCTL0.PERITV[4:0]). The expected value is configured in Periodic transaction expected and masked value bits (CDCTL1.PEREXP[31:0] and CDCTL2.PERMSK[31:0]). Table 35.6 shows the configured register bits for periodic manual-command. The operating flow is illustrated in Figure 35.26.

**Table 35.6 Manual-command configuration for periodic mode**

Transaction	Transaction type	Command	Command size	Address	Address size	Data (x = 0, 1)	Data size	Latency cycle
Read Transaction	CDTBUF0. TRTYPE	CDTBUF0. CMD[15:0]	CDTBUF0. CMDSIZE[1: 0]	CDABUF0. ADD[31:0]	CDTBUF0. ADD SIZE[2:0 ]	CDDxBUF0. DATA[31:0]	CDTBUF0. DATASIZE[3: 0]	CDTBUF0. LATE[4:0]

### 35.3.3 Memory-mapping

This section describes the memory-mapping mode. This mode automatically converts system bus access for pre-configured memory area into xSPI transaction.

#### 35.3.3.1 Configuration

In this operation, the payload of address and data field are delivered from system bus signals. The information of command field and size are delivered from the configured register bits. [Table 35.7](#) shows the register bits configured for memory-mapping.

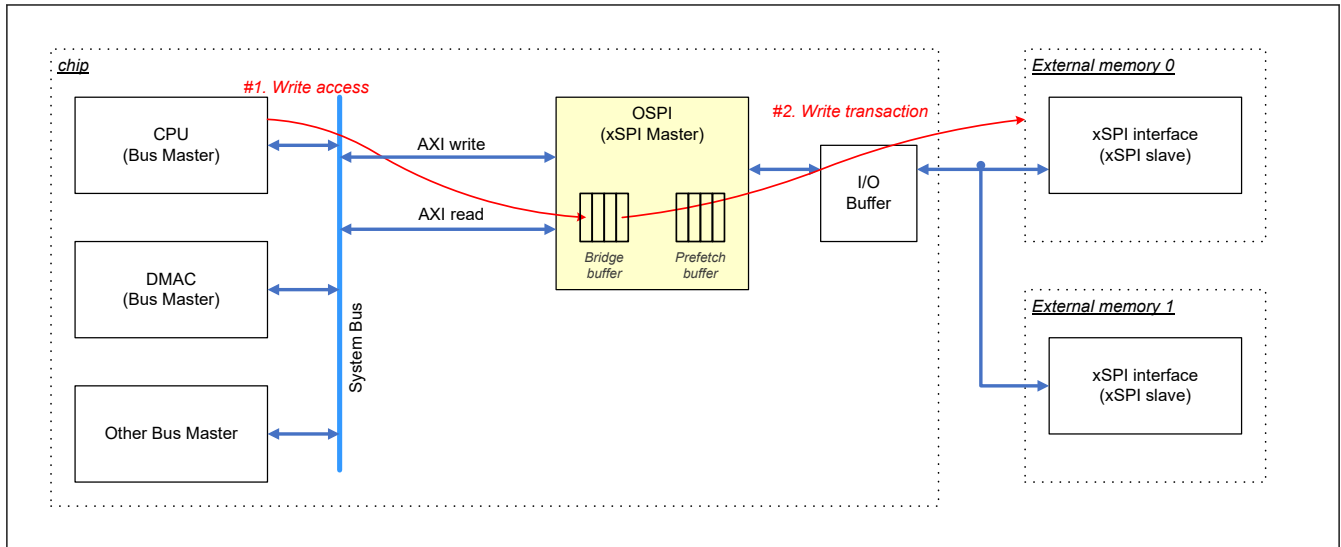
**Table 35.7 Memory-mapping configuration for memory area access (n = 0, 1)**

System bus Transaction	Format Change mode	Command	Command size	Address	Address size	Data	Data size	Latency cycle
Write for slave n memory area	Normal	CMCFG2CSn. WRCMD[15:8]	1 byte	SAWADDR[x:0]	CMCFG0C Sn. ADD SIZE[ 1:0]	SWDATA	Up to SAWLEN and SAWSIZE	CMCFG2CSn. WRLATE[4:0]
	8D-8D-8D profile 1.0	CMCFG2CSn. WRCMD[15:0]	2 bytes					
	8D-8D-8D profile 2.0 Command Modifier	CMCFG2CSn. WRCMD[15:8]	1 byte	{SAWADDR[27:4], 0000000000000b, SAWADDR[3:1]}	5 bytes			
	8D-8D-8D profile 2.0 Extended Command Modifier	CMCFG2CSn. WRCMD[15:13]	3 bits	{0b, SAWADDR[31:4], 0000000000000b, SAWADDR[3:1]}	45 bits			
Read for slave n memory area	Normal	CMCFG1CSn. RDCMD[15:8]	1 byte	SARADDR[x:0]	CMCFG0C Sn. ADD SIZE[ 1:0]	SRDATA	Up to SARLEN and SAR SIZE	CMCFG1CSn. RDLATE[4:0]
	8D-8D-8D profile 1.0	CMCFG1CSn. RDCMD[15:0]	2 bytes					
	8D-8D-8D profile 2.0 Command Modifier	CMCFG1CSn. RDCMD[15:8]	1 byte	{SARADDR[27:4], 0000000000000b, SARADDR[3:1]}	5 bytes			
	8D-8D-8D profile 2.0 Extended Command Modifier	CMCFG1CSn. RDCMD[15:13]	3 bits	{0b, SARADDR[31:4], 0000000000000b, SARADDR[3:1]}	45 bits			

Note: The MSByte of Address can be replaced with Address Replace Enable and Code bits (CMCFG0CSn.ADDRPEN[7:0] / ADDRPCD[7:0]).

#### 35.3.3.2 Write Access Operation

At accepting write access for memory area from system bus, this xSPI Master stores all payload data in internal bridge buffer and then issues a write transaction into xSPI slave. [Figure 35.12](#) shows the operation summary.



**Figure 35.12 Write access for memory area**

The operation of xSPI bus changes depending on system bus's burst type. When the burst type is single type or incremental type, one system bus transaction triggers for one xSPI frame. When the burst type is wrap type and the CMCFG0CSn.WPBSTMD is 0, one system bus transaction triggers two xSPI frames. [Figure 35.13](#) shows the relationship between AXI and xSPI frames.

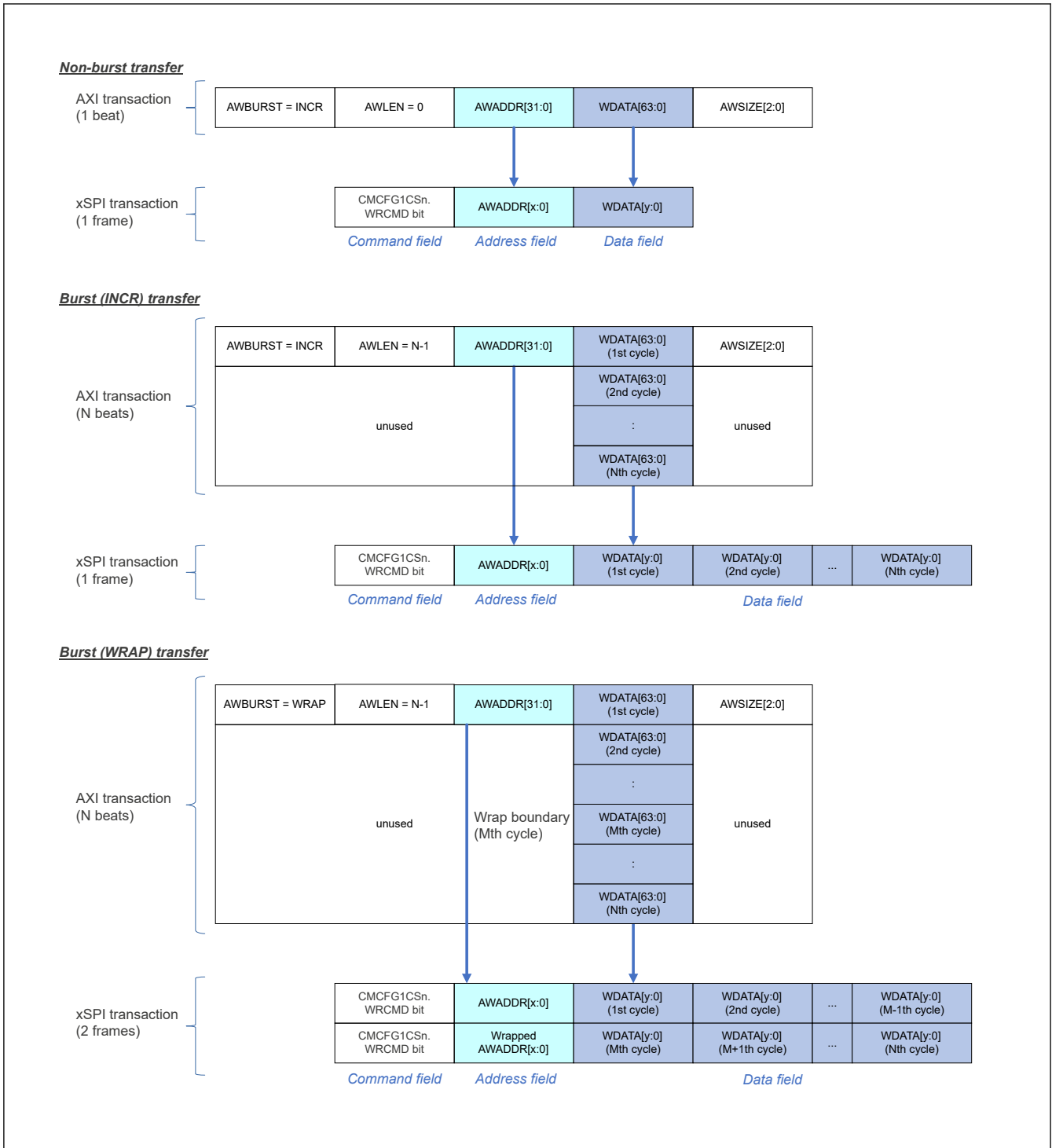


Figure 35.13 xSPI frame format in write access (Normal format)

### 35.3.3.3 Combination Function

At the system bus write access for memory area, this xSPI master has the function to combine the write data for high throughput on xSPI bus. When this function is enabled (BMCFGCHn.MWRCOMB = 1), this xSPI master transmits a xSPI frame with the selected size while the sequential address is incremental\*1. When one of the below conditions is detected, even though not reaching to the target size (BMCFGCHn.MWRSIZE[7:0]), this xSPI master transmits the pending data into xSPI bus.

- Non-incremental address is detected.
- Different burst type is detected.

- Read transaction is detected.
- Access for different slave is detected.
- Memory Write Data Push bit (BMCTL1.MWRPUSHCHn (n = 0, 1)) is set.

This function could be useful for any slave device to request a chunk of data at a time. In the case, system bus master shall continue to provide the fixed data size with incremental address. e.g. there is any device to request to write in page unit.

Figure 35.14 shows the operation when enable the combination function.

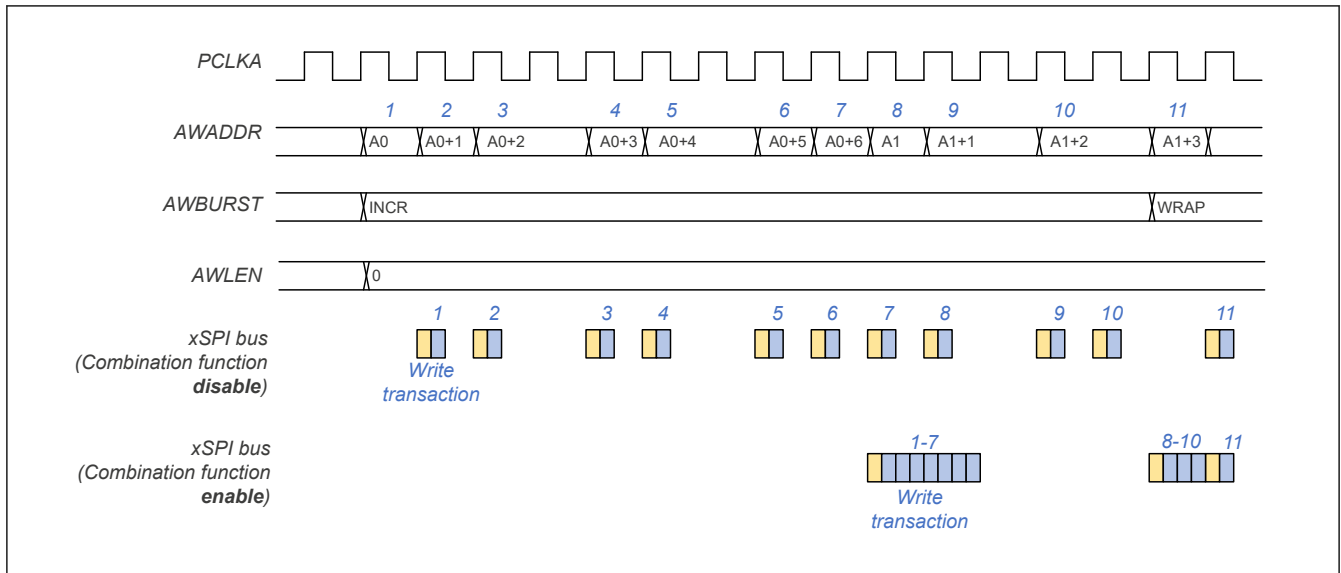


Figure 35.14 Combination function

Note 1. The access which comply all below condition is considered as incremental address.

- Transaction type is INCR.
- The access's start address is continuous to the previous last write address.
  - The access's start position of WSTRB is treated as start address.
  - Previous access's last WSTRB is treated last write address.

Figure 35.15 shows data combined example with AXI access.

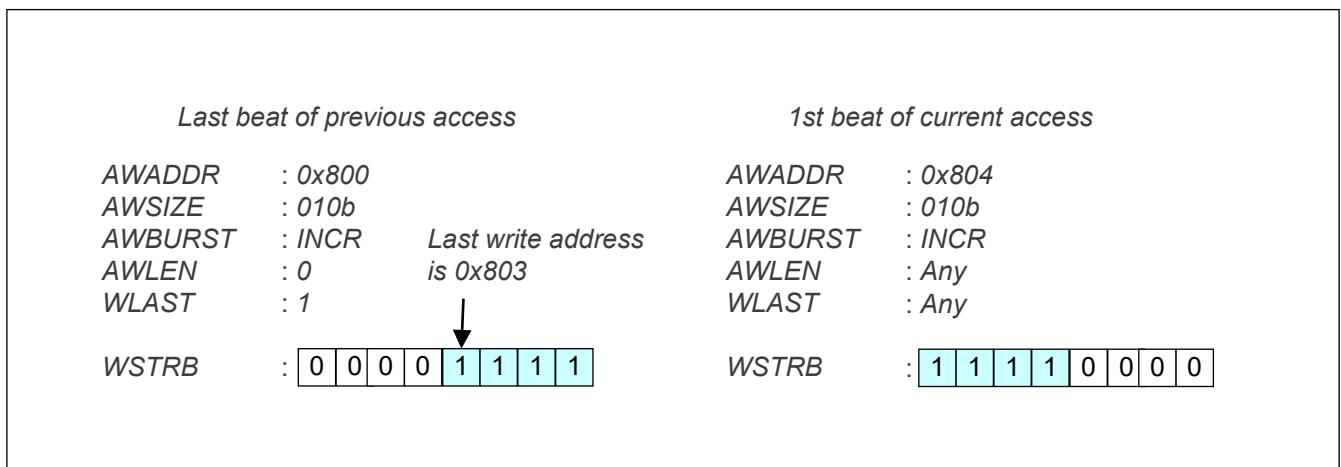
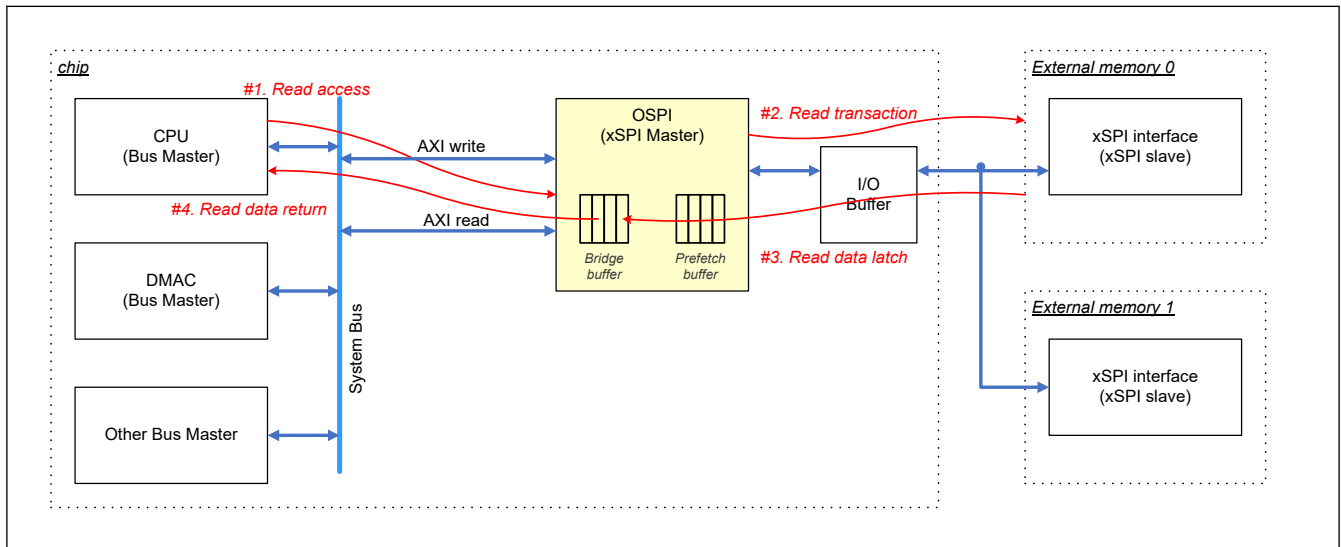


Figure 35.15 Data combined example with AXI access

### 35.3.3.4 Read Access Operation

At the read access for memory area, soon after detected the read access, this xSPI Master issues a read transaction into xSPI slave. [Figure 35.16](#) shows the operation summary.



**Figure 35.16 Read access for memory area**

The operation of xSPI bus changes depending on burst type. When the type is single or increment type, one system bus's read transaction triggers one xSPI frame. When the type is wrap type and the CMCFG0CSn.WPBSTMD is 0, one system bus read transactions triggers two xSPI frames. [Figure 35.17](#) shows the relationship between AXI and xSPI frames.

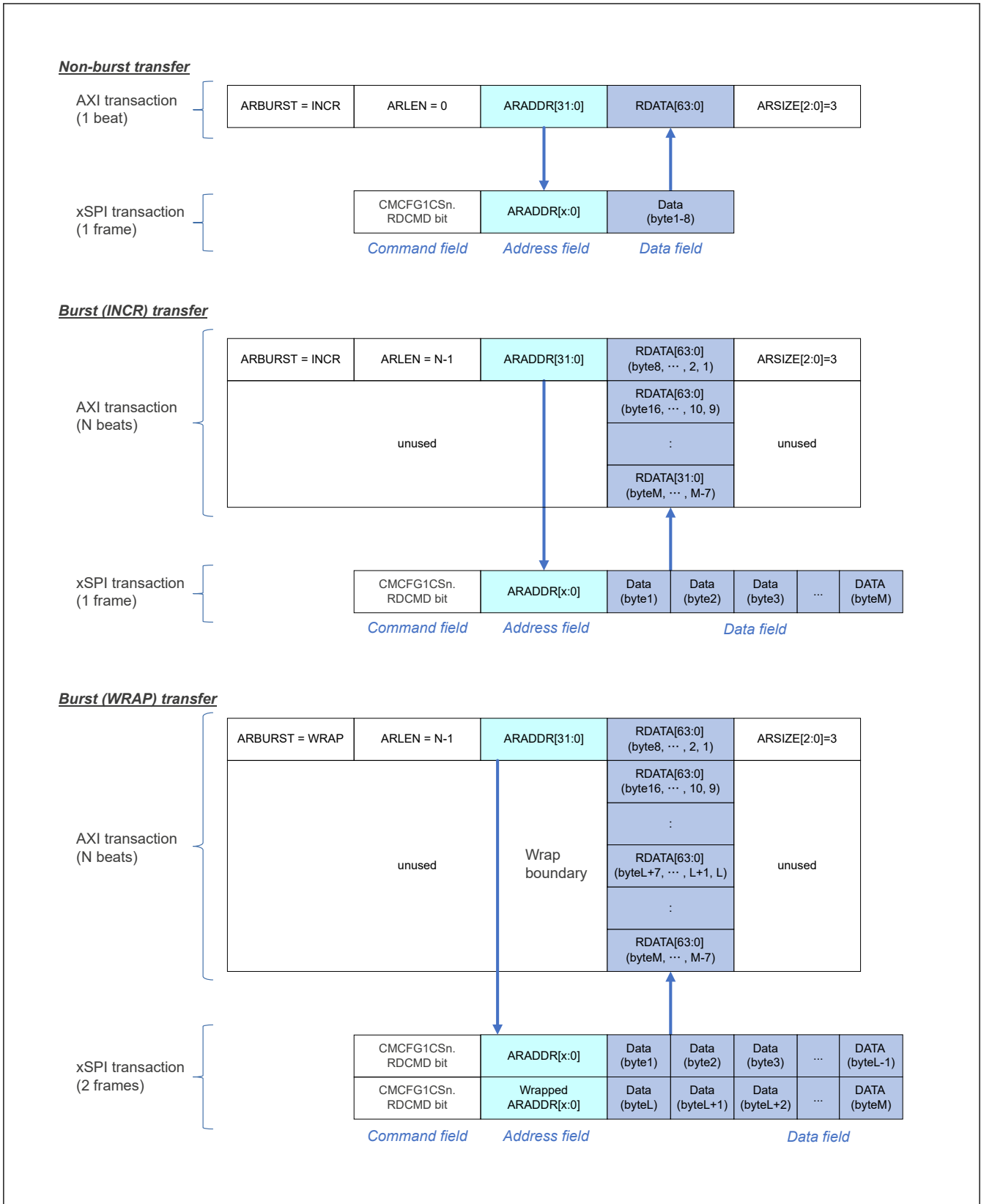


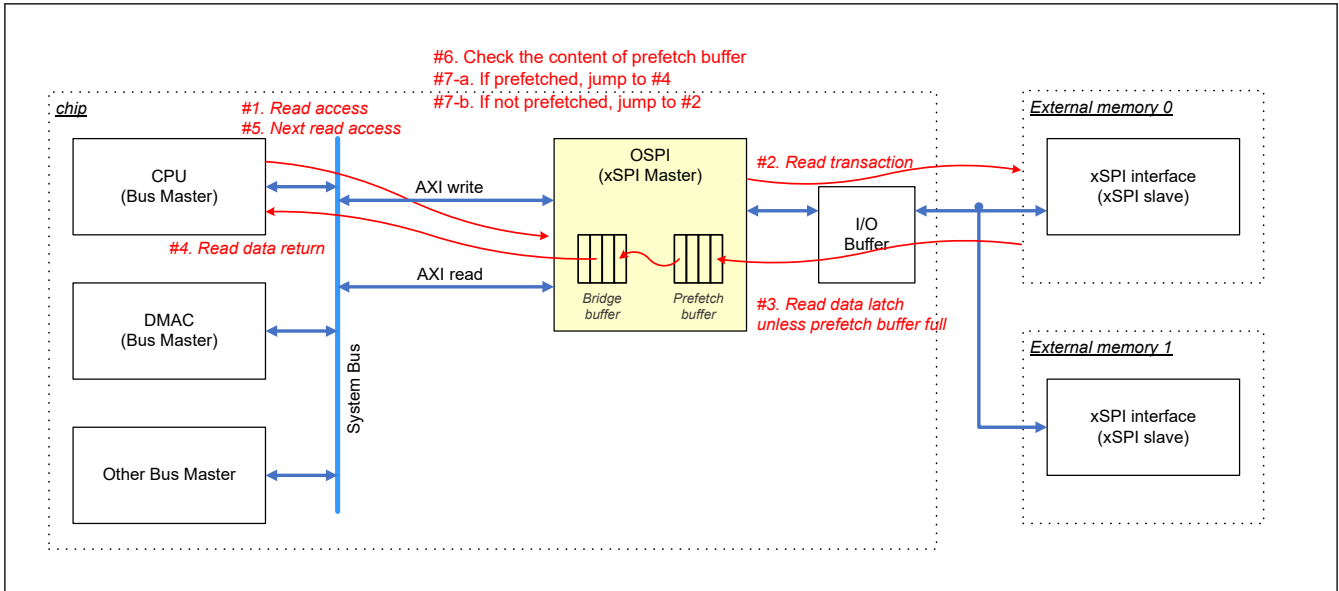
Figure 35.17 xSPI frame format in read access (Normal format)

### 35.3.3.5 Prefetch Function

At the read access for memory area from system bus, this xSPI Master has the function to prefetch the read data for reducing the latency. When enabled this function (BMCFGCHn.PREEN = 1), this xSPI Master continues to read the



incremental address and store the read data from xSPI slave in internal prefetch buffer. And this xSPI master searches in prefetch buffer for the following read access from system bus. If found the target read data in prefetch buffer, this xSPI Master returns the data from prefetch buffer. If not found, this xSPI Master clears prefetch buffer and newly issues a read transaction into xSPI slave. This function is effective in application such as the consecutive read addresses are close. But it is not effective in application such as the consecutive read addresses are not incremental because xSPI read frame for prefetch uses xSPI bus. [Figure 35.18](#) shows the operation summary.



**Figure 35.18 Read access for memory area with prefetch enabled**

- Note: When enabled this prefetch function, Bus Master could read from not a slave device but the internal prefetch buffer. When accessed to the same address from multiple Bus Masters, this xSPI Master does not guarantee to read the latest data. If Bus Master wish to read the latest data from a slave device, it should read after cleared the prefetch buffer (BMCTL1.PBUFCLRCHn (n = 0, 1)).
- Note: Prefetch buffer is implemented as FIFO-based, and when read access is issued, the data before the access address is discarded from the buffer. And when next access is issued to the region which is discarded at previous access, this module issues the xSPI read access again to fill the prefetch buffer.
- Note: OSPI has the 1line buffer which keeps the last 8Byte read data from prefetch buffer. When read access is issued to the same address to the data in 1line buffer, OSPI returns read data from 1line buffer.

### 35.3.3.6 XiP Mode

Some slave devices have a mode (XiP mode) in which the command phase is not required for lower latency. While in this mode, the xSPI master skips sending the command and the slave device implicitly performs the command that was executed in the previous transaction. When enabled XiP mode bit (CMCTLCHn.XIPEN = 1), this xSPI master inserts XiP enter code (CMCTLCHn.XIPENCODE[7:0]) in latency field. When disabled XiP mode bit (CMCTLCHn.XIPEN = 0), this xSPI master inserts XiP exit code (CMCTLCHn.XIPEXCODE[7:0]) in latency field. This function is available only for memory-mapping mode.

And when this xSPI master transmits XiP disable pattern, this master clears XiP mode bit and disables XiP mode configured for both channels. Note that it is not possible to disable XiP mode for only one channel by transmitting XiP disable pattern.

- Note: When enough latency cycle does not exist for XiP code, this xSPI Master could not insert XiP code.
- Note: XiP mode could be used only for unidirectional access to a slave. The write transaction and read transaction should be separated.
- Note: The XiP exit code is inserted once when disabled. More details, See [Figure 35.30](#).

### 35.3.4 Pattern Control

This xSPI Master has the function to transmit 3 type of patterns which is not xSPI frame format. The pattern is triggered by setting trigger bit (LPCTL0-1.PATREQ).

#### 35.3.4.1 XiP Disable Pattern

XiP Disable pattern transmits any pattern with the configured length and value (LPCTL0.XD1LEN[4:0] / XD1VAL / XD2LEN[4:0] / XD2VAL). It uses OM\_SCLK, OM\_SIO7-0 signals. The number of output pin can be configured by XiP Disable pattern pin bits (LPCTL0.XDPIN[1:0]). It may be used to disable XiP mode for legacy SPI. Figure 35.19 shows the timing-chart.

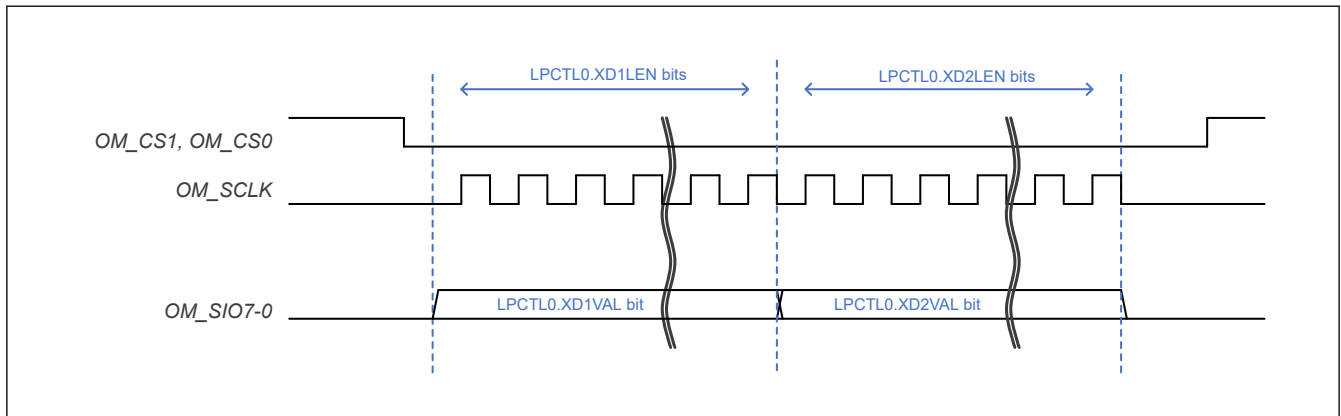


Figure 35.19 XiP Disable pattern

#### 35.3.4.2 Reset Pattern

Reset pattern transmits the pattern specified in Serial Flash Reset Signaling Protocol. Figure 35.20 shows the timing chart. CS Low/High width is configured with Reset Pattern Length bits (LPCTL1.RSTWID[2:0]). xSPI slave will sample the data input at the rising edge of CS. Setup time for data output is configured with Reset pattern data output setup time bits (LPCTL1.RSTSU[2:0]). The setup time should be less than Reset pattern width always.

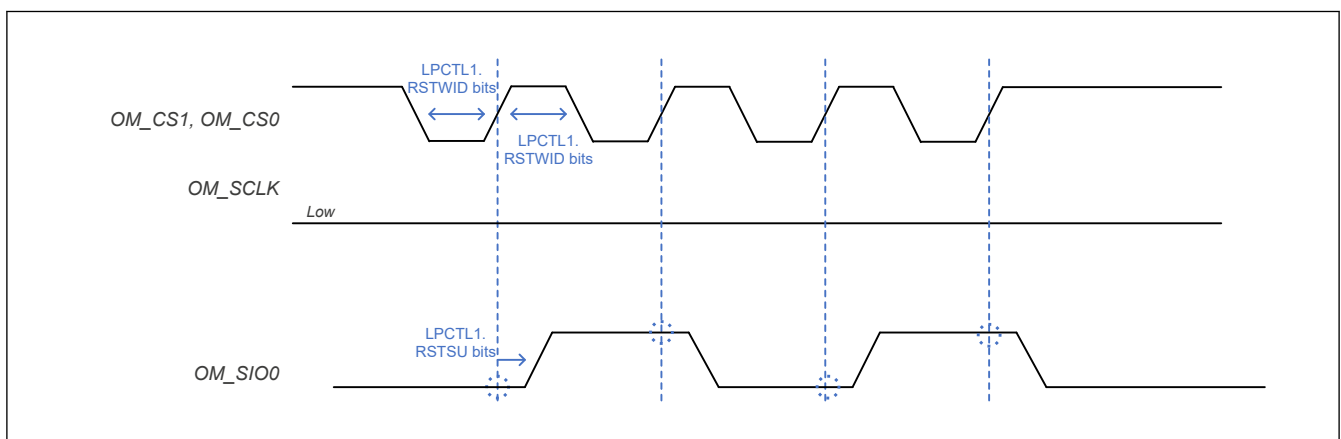


Figure 35.20 Reset pattern

Note: In the protocol, CS Low/High width is defined as minimum 500 ns and Setup time is defined as minimum 6 ns.

#### 35.3.4.3 CS Only Pattern

CS Only pattern activates CS port with the configured length bits (LPCTL1.RSTWID[2:0]). It may be used to resume from Deep Power Down state. Figure 35.21 shows the timing-chart.

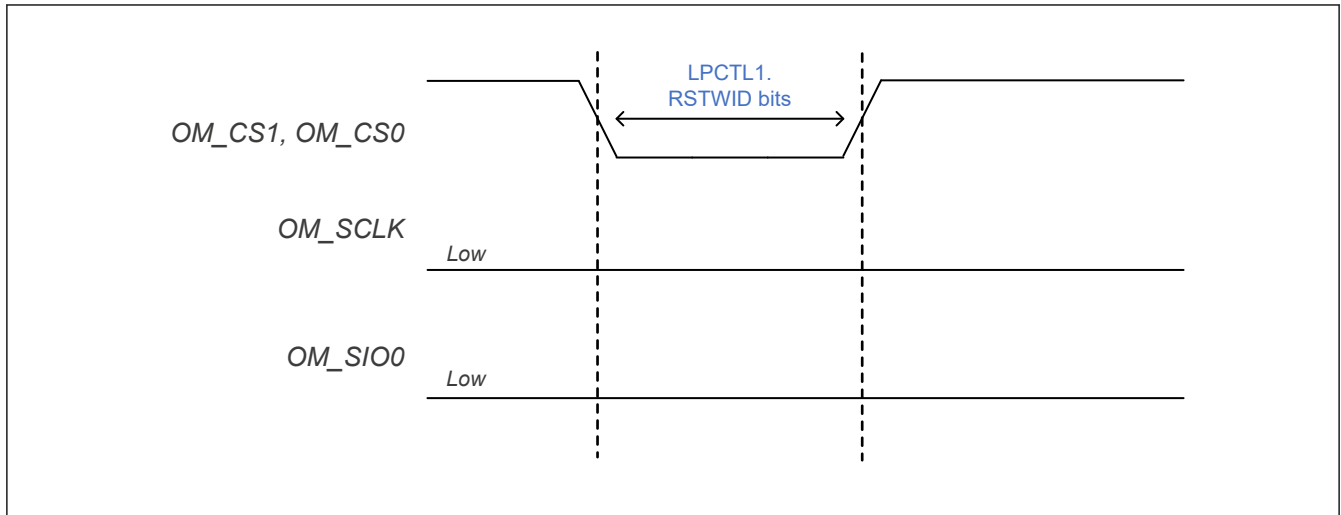


Figure 35.21 CS Only pattern

### 35.3.5 Integrity Checking

This xSPI Master can detect some errors. Table 35.8 shows the error list and the detail behavior.

Table 35.8 Error list (n = 0, 1)

Error type	Event	Flag bit	Note (action)
Calibration failed	When the read data did not match the expected value during automatic calibration.	INTS.CAFAILCSn	It results in writing unexpected data to xSPI slave.
System bus error	When an error response occurred on AXI slave interface for memory-mapping.	INTS.BUSERRCHn	This xSPI master shall be reset for fatal error.
ECC error detection	When detected the falling edge on <i>OM_ECSINT1</i> port. It can be useful only for xSPI slave with ECC detection function.	INTS.ECSCS1	Only notify the error event of xSPI slave.
<i>OM_DQS</i> timeout	When <i>OM_DQS</i> does not toggle in read transaction with using <i>OM_DQS</i> .	INTS.DSTOCSn	Both xSPI master and xSPI slave should be reset for fatal error.
Periodic transaction timeout	When the read value does not match with the expected value in periodic manual-command mode.	INTS.PERTO	Depending on the status of function.

### 35.3.6 Interrupts

This xSPI Master has an interrupt port.

It can monitor with Interrupt Status Register (INTS). In case of initialization phase, it can be programmable with Interrupt Enable register (INTE). Table 35.9 shows OSPI interrupt sources, and Table 35.10 shows the related register bit.

Note: Interrupt pulse port signal is not asserted when the corresponding bit of INTE is set after the interrupt event is detected.

Table 35.9 OSPI interrupt sources

Name	Interrupt sources	DMAC activation
OSPI0_ERR	Error	Not possible
OSPI0_CMP	Complete	Not possible

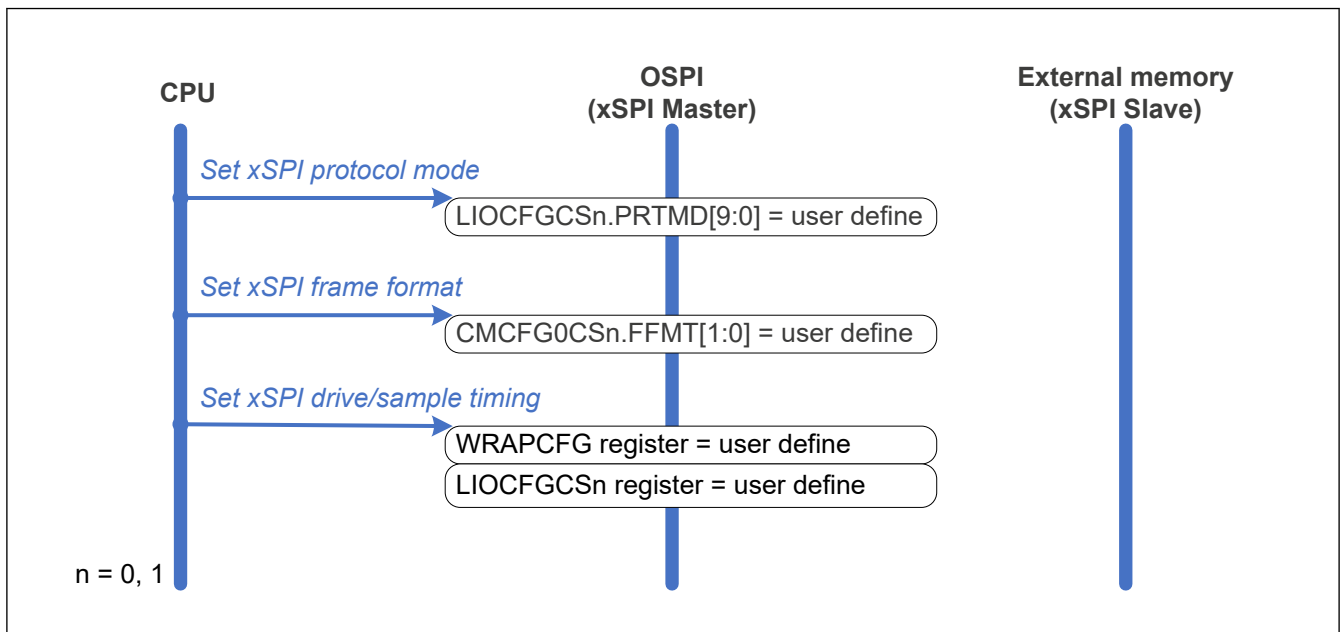
**Table 35.10 Interrupt register bit**

Flag bit	Enable bit	Clear bit	Interrupt sources
CASUCCS1	CASUCCS1E	CASUCCS1C	OSPI0_CMP
CASUCCS0	CASUCCS0E	CASUCCS0C	
CAFAILCS1	CAFAILCS1E	CAFAILCS1C	OSPI0_ERR
CAFAILCS0	CAFAILCS0E	CAFAILCS0C	
BUSERRCH0	BUSERRCH0E	BUSERRCH0C	OSPI0_ERR
INTCS1	INTCS1E	INTCS1C	OSPI0_ERR
ECSCS1	ECSCS1E	ECSCS1C	OSPI0_ERR
DSTOCS1	DSTOCS1E	DSTOCS1C	OSPI0_ERR
DSTOCS0	DSTOCS0E	DSTOCS0C	
PERTO	PERTOE	PERTOC	OSPI0_ERR
PATCMP	PATCMPE	PATCMPC	OSPI0_CMP
CMDCMP	CMDCMPE	CMDCMPC	OSPI0_CMP

### 35.3.7 Flows of Operations

#### 35.3.7.1 Flow of Configuration

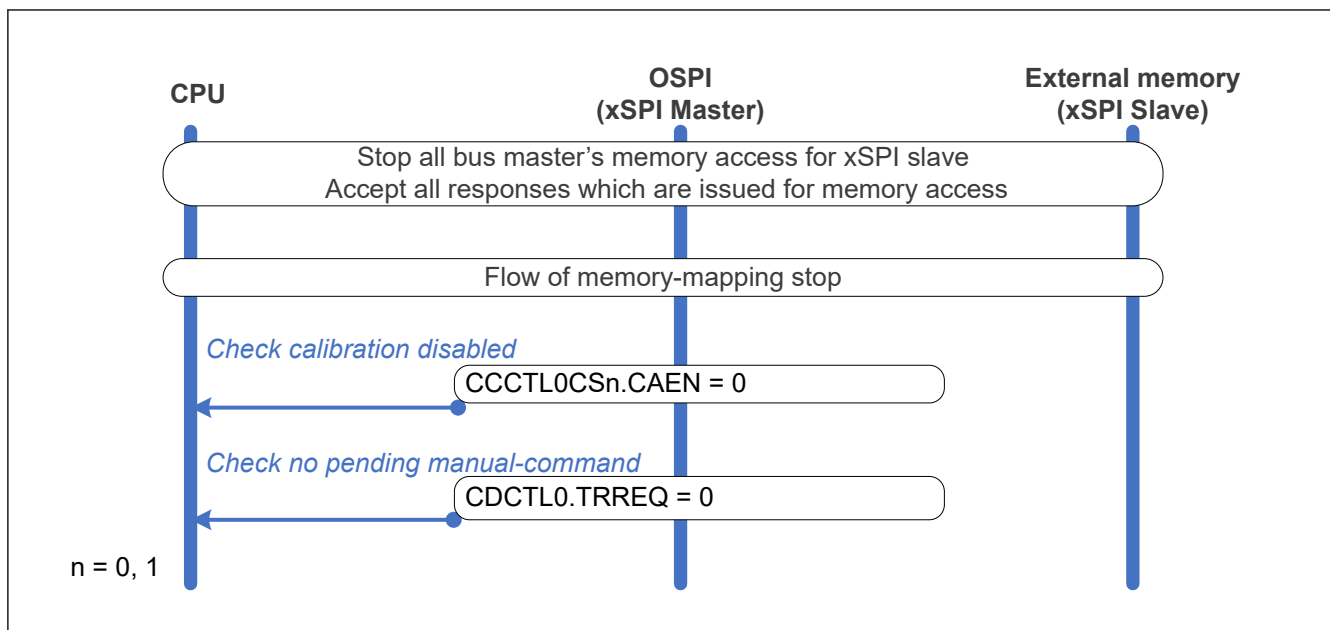
Figure 35.22 shows flow of configuration.



**Figure 35.22 Flow of configuration**

#### 35.3.7.2 Flow of Communication Stop

Figure 35.23 shows flow of communication stop.



**Figure 35.23 Flow of communication stop**

**Note:** In case of re-config of any configuration register, all communication with xSPI slave shall be stopped surely to avoid race condition between register setting and memory access. It means that the automatic calibration is disabled, and there is no pending manual-command and memory-mapping access.

### 35.3.7.3 Flow of Automatic Calibration

Figure 35.24 shows flow of automatic calibration.

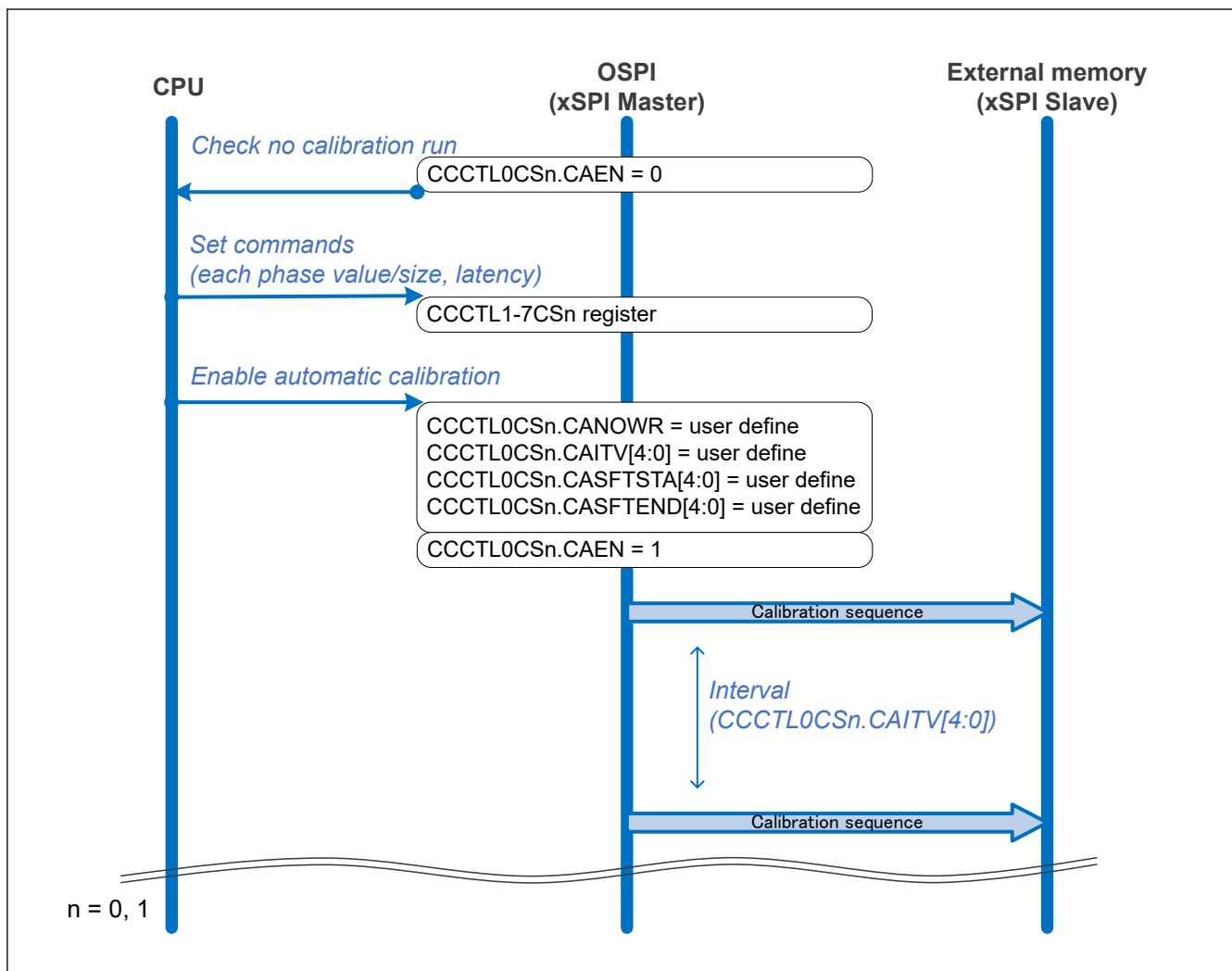


Figure 35.24 Flow of automatic calibration

### 35.3.7.4 Flow of Manual-command Procedure

Figure 35.25 shows manual-command procedure for direct mode.

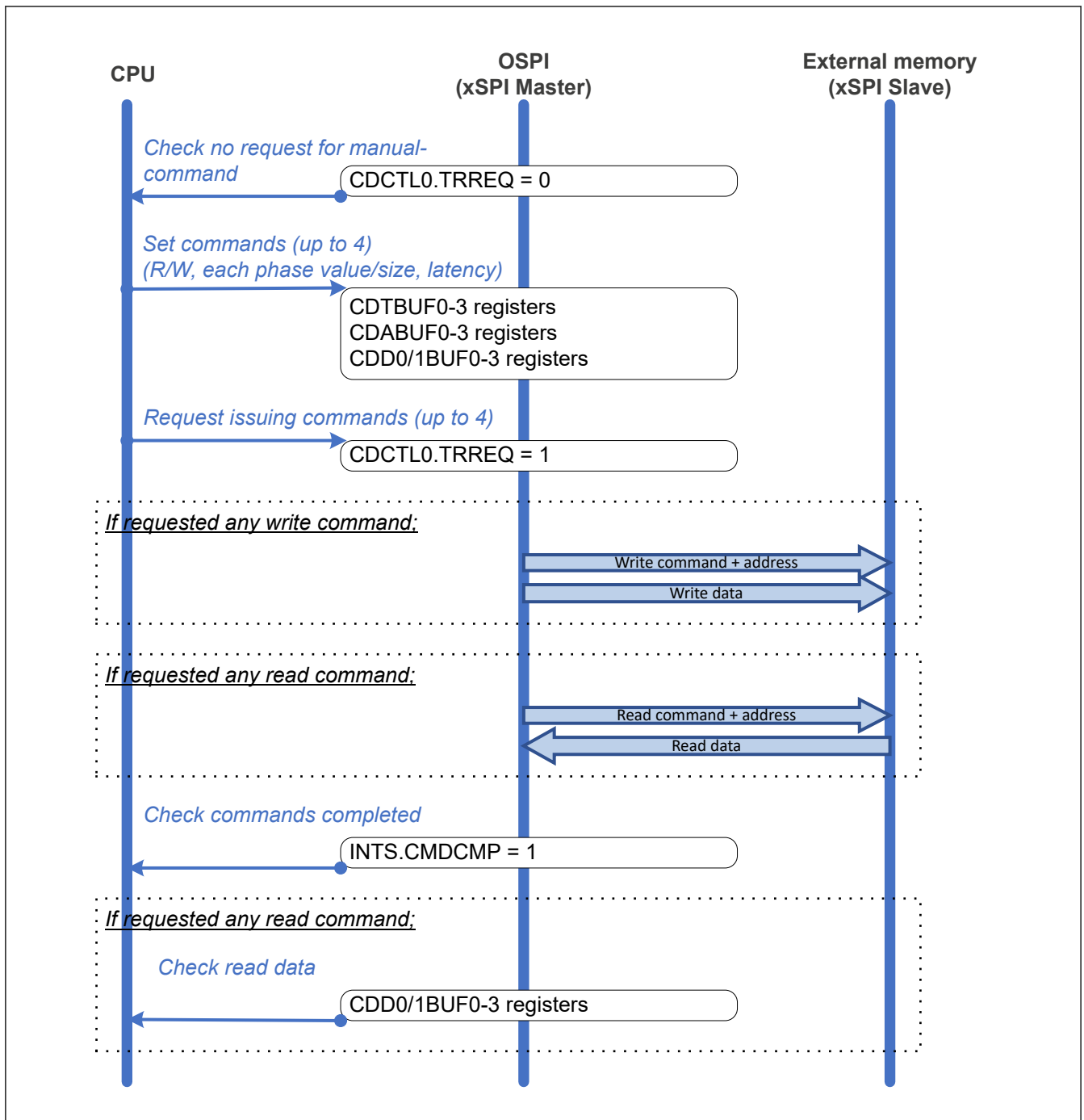


Figure 35.25 Flow of manual-command procedure for direct mode

Figure 35.26 shows manual-command procedure for periodic mode.

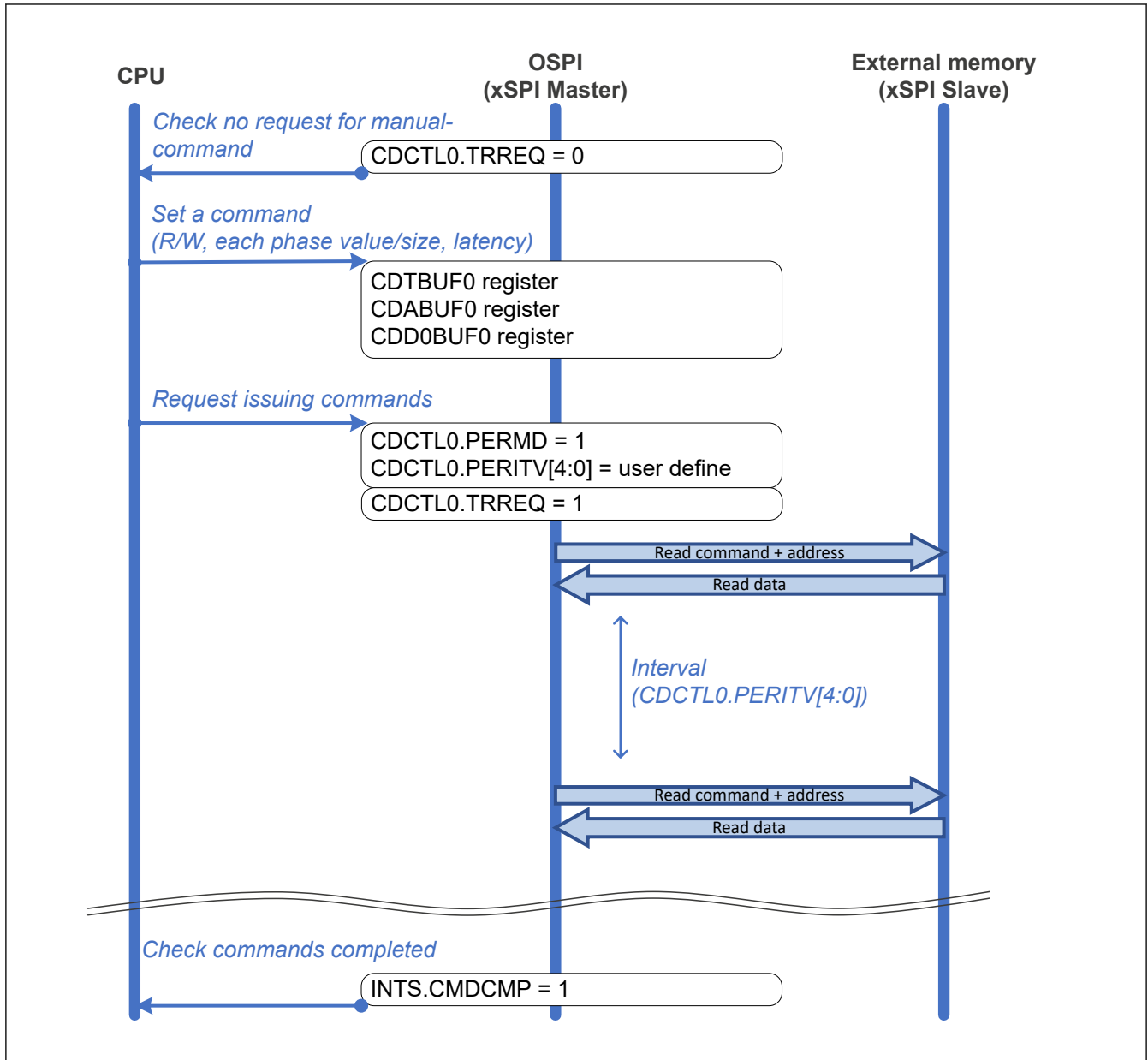


Figure 35.26 Flow of manual-command procedure for periodic mode

### 35.3.7.5 Flow of Memory-mapping

Figure 35.27 shows flow of memory-mapping.



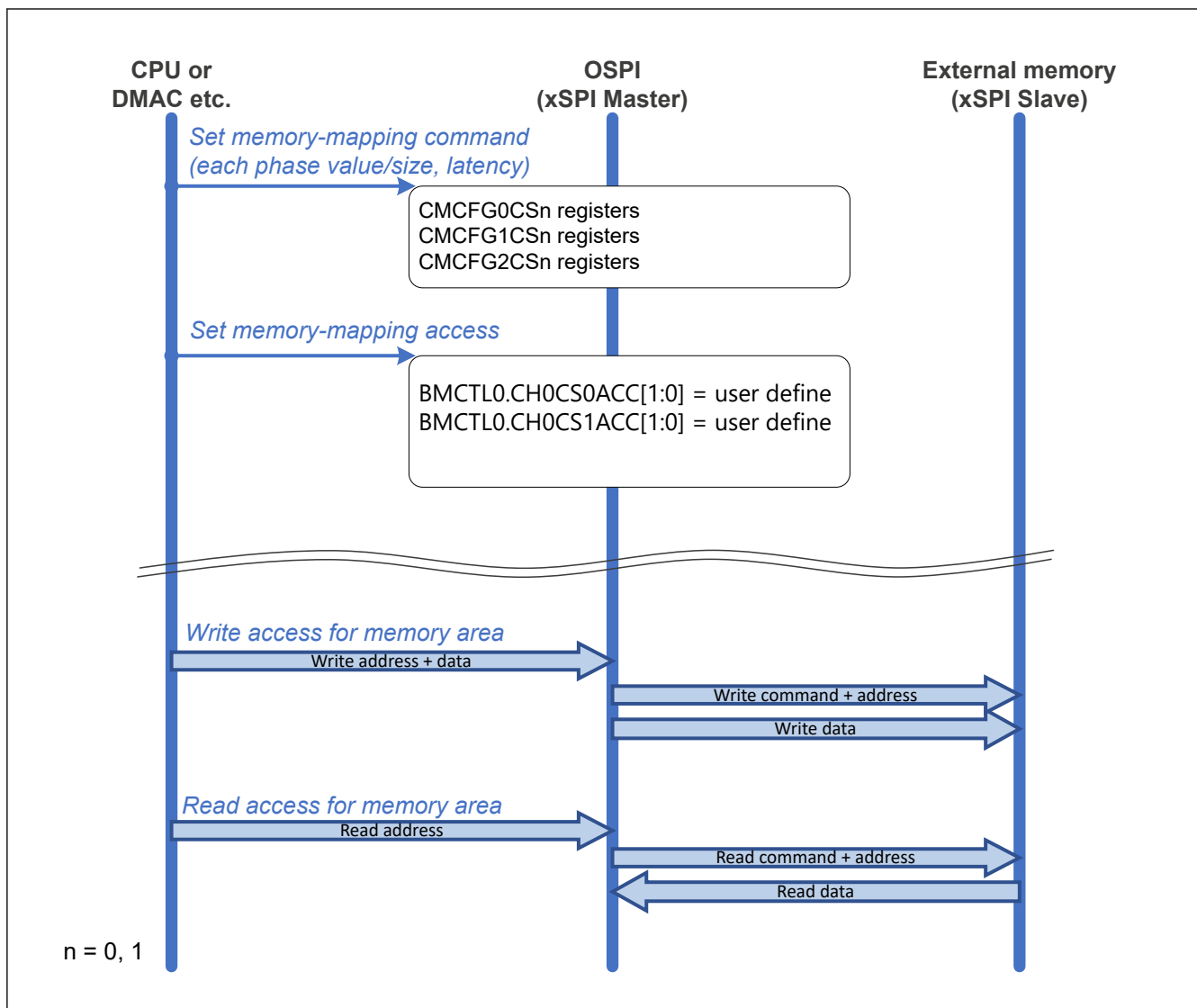


Figure 35.27 Flow of memory-mapping

### 35.3.7.6 Flow of Memory-mapping Stop

Figure 35.28 shows flow of memory-mapping stop.

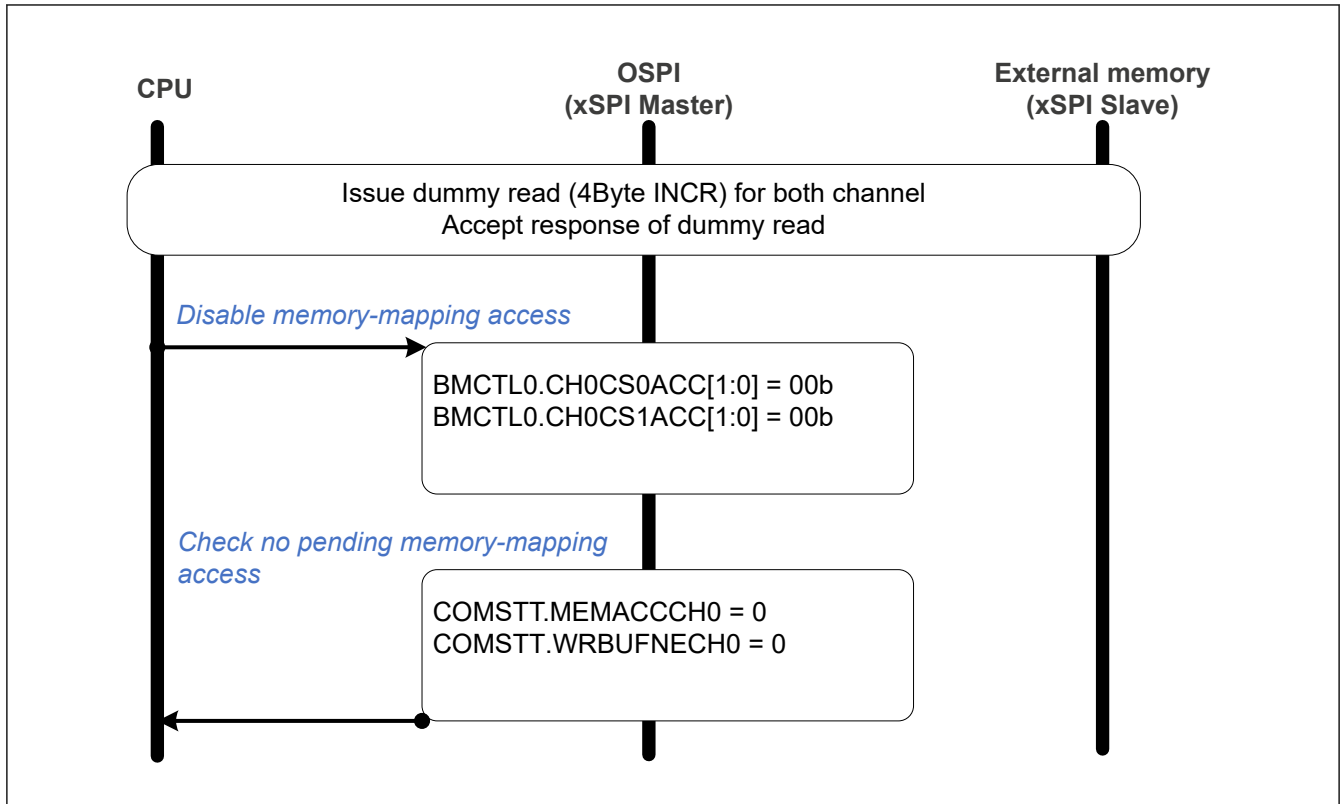


Figure 35.28 Flow of memory-mapping stop

### 35.3.7.7 Flow of Pattern Request

Figure 35.29 shows flow of pattern request. Before requesting any pattern, any ongoing commands should be completed or canceled.

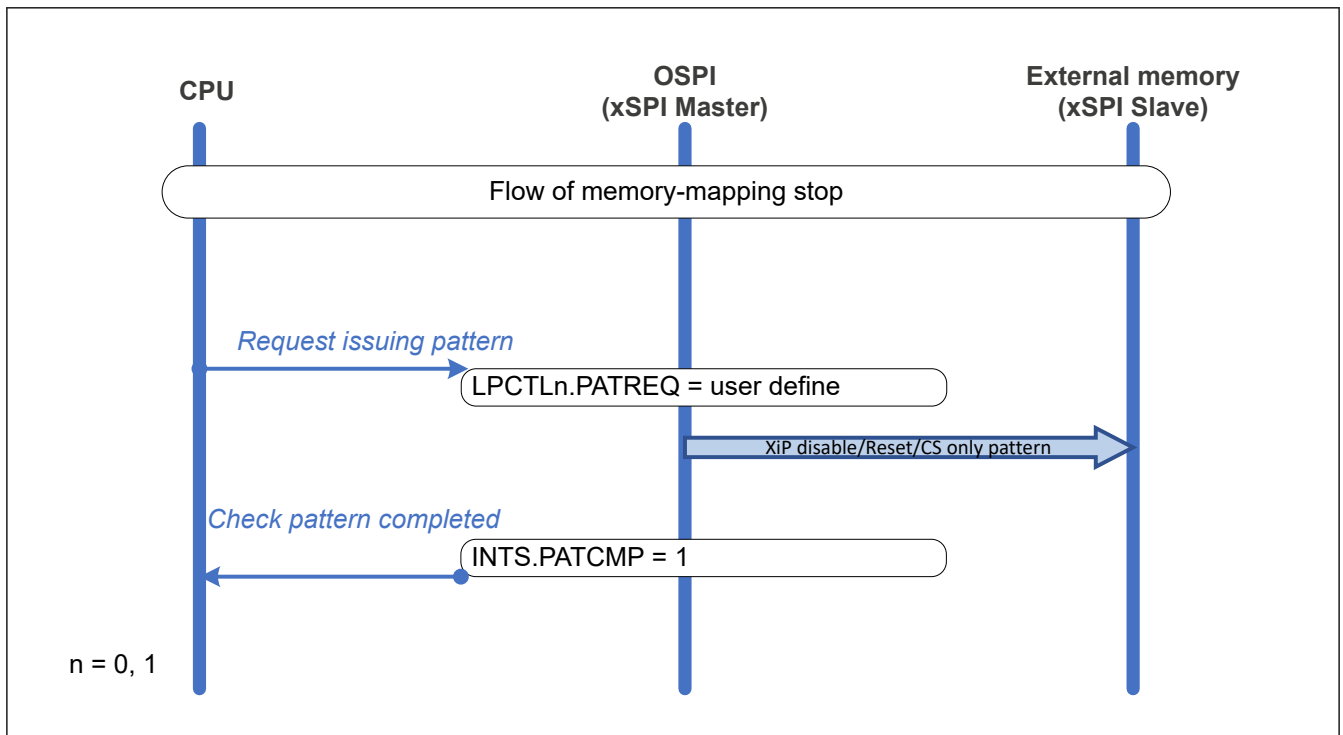


Figure 35.29 Flow of pattern request

### 35.3.7.8 Flow of XiP Mode

Figure 35.30 shows flow of XiP mode enable/disable.

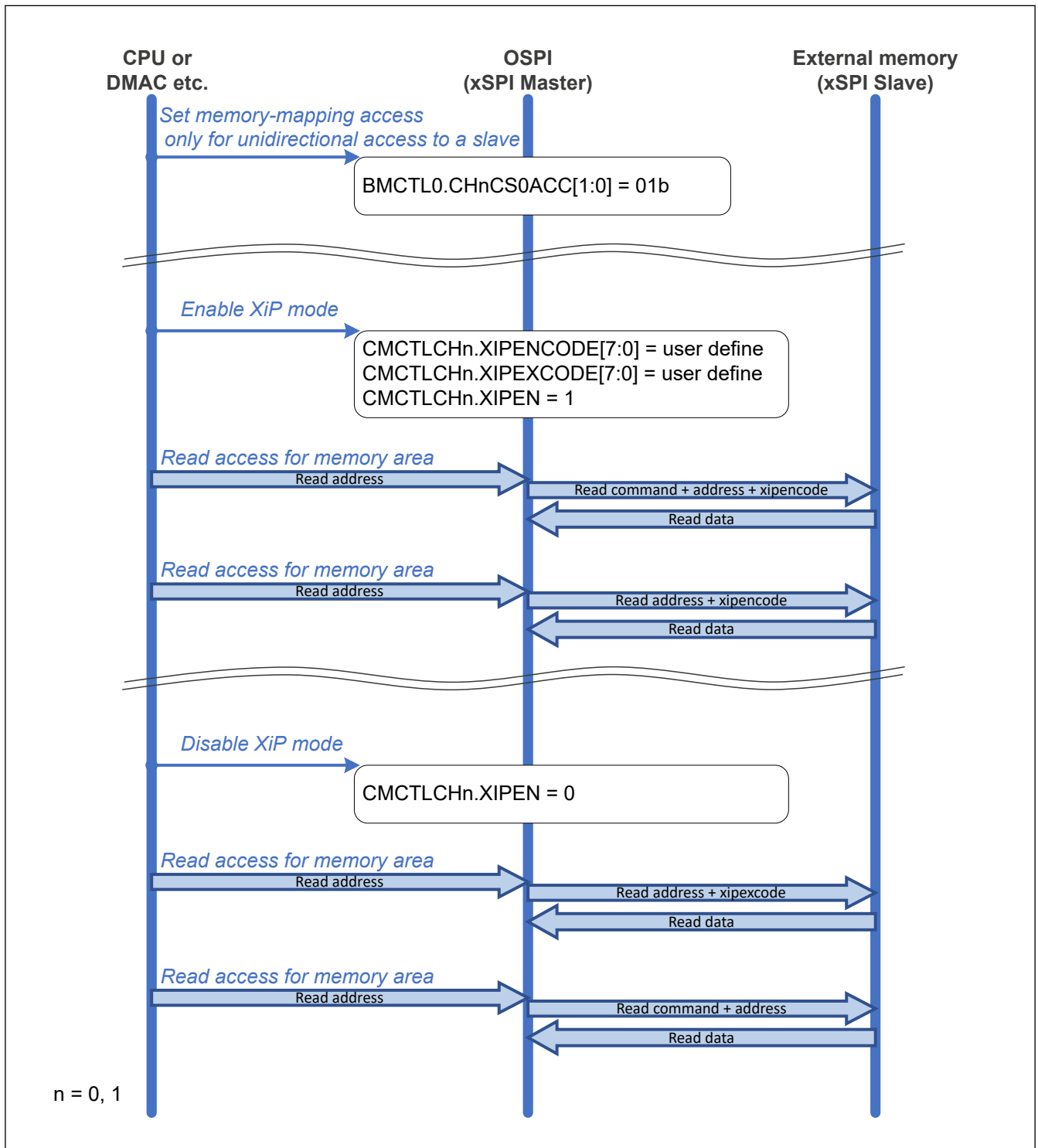


Figure 35.30 Flow of XiP mode enable/disable

### 35.3.7.9 Flow of Read While Write

Figure 35.31 shows example flow of read while write.

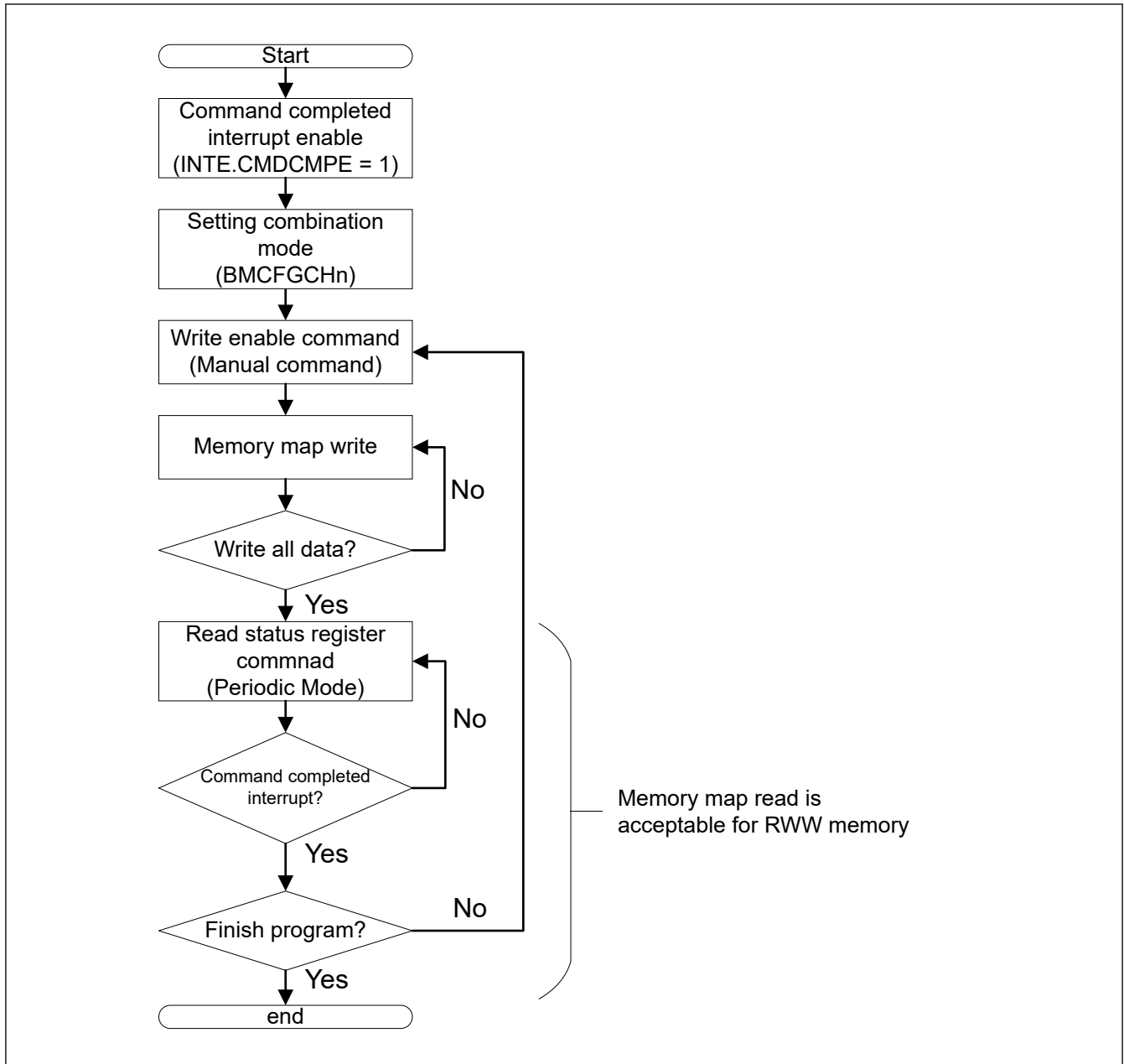


Figure 35.31 Flow of read while write

### 35.3.8 Usage Notes

#### 35.3.8.1 Constraint on Memory Access

The memory access has the following constraints:

- Prohibit to access last 128 byte area of memory support area.
- Not using prefetch function if application access to the last 128 byte area

#### 35.3.8.2 Constraint on Burst Length

AXI bus master must have a Burst Length of 16 or less.

#### 35.3.8.3 Memory Write Combination Mode

When BMCFGCHn.MWRCOMB is 1, memory write combination mode is enabled. However, combination does not work if the bus master is CPU, and OSPI data is under 32bit. Table 35.11 shows the possibility of data write for each bus master.

**Table 35.11 Data write possibility for each bus master**

Bus Master	Combination Enable	Combination Disable
CPU under 64 bit Access	Not possible	Possible
CPU 64 bit Access	Possible	Possible
DMAC/DTC	Possible	Possible
EDMAC	Possible	Possible
CEU	Possible	Possible

### 35.3.8.4 Module-stop function

OSPI operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The OSPI module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

### 35.3.8.5 Restriction in 8D-8D-8D profile 1.0 format

"When read or write access to the Octal Serial Peripheral Interface (OSPI) is performed from the following bus masters with 8-bit wide, access to odd addresses is prohibited as it will not be worked correctly. See [Table 35.12](#) for the applicable conditions and their workarounds for each bus master"

**Table 35.12 Workaround for the restriction in 8D-8D-8D profile 1.0 format**

Bus master		Conditions that require workaround	Workaround
CPU	Normal memory	Debugger is connected	Access only even addresses.
	Device memory	Always	Access only even addresses.
DMAC/DTC	DMAC	DMTMD.SZ[1:0] = 00b	Access only even addresses.
	DTC	MRA.SZ[1:0] = 00b	Access only even addresses.

### 35.3.8.6 Byte data order in 8pin-DDR mode

In 8pin-DDR mode, OSPI read and write data in the order of byte0 at rising edge and byte1 at falling edge.

Some memory devices read and write in the swapped order in 8pin-DDR mode. For these devices, if 8pin-DDR read the data which is written in other than 8pin-DDR mode, this data should be written in swapped order.

## 36. Decryption On The Fly (DOTF)

All operations of DOTF are not guaranteed.

### 36.1 Overview

DOTF has a function to decode read data of the AXI bus using AES core.

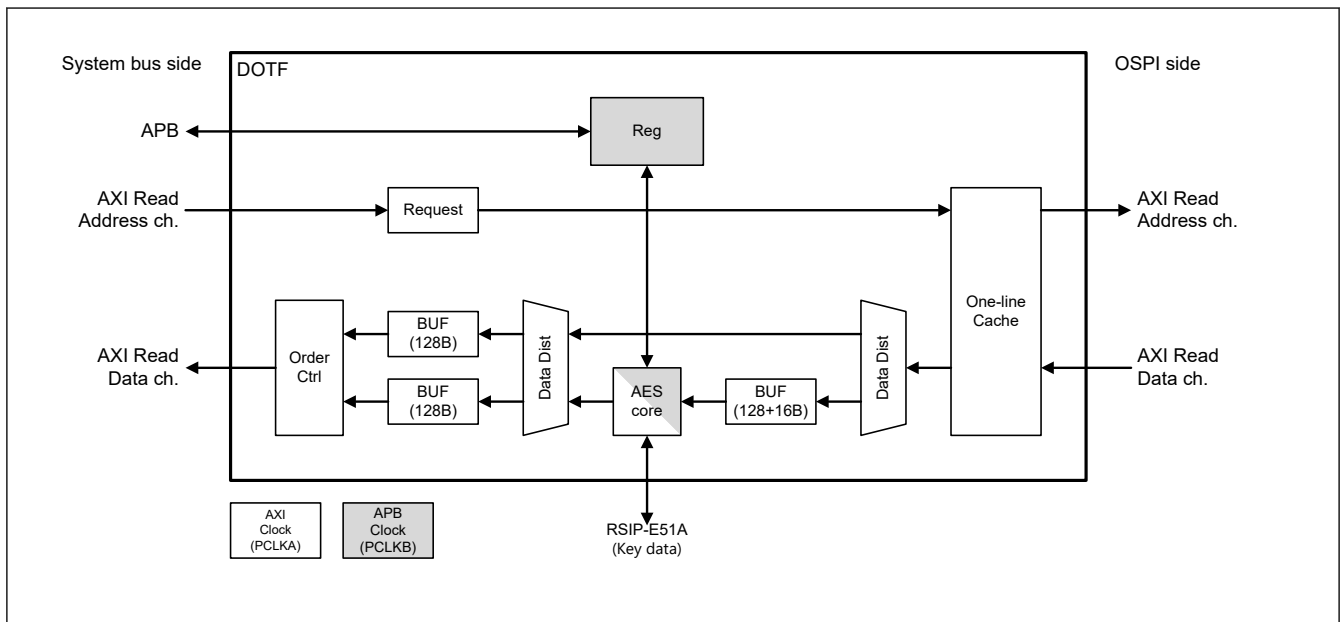
Table 36.1 lists the DOTF specifications and for a system using DOTF, see section 35.1. Overview.

**Table 36.1 DOTF Specification**

Item	Description
Clock Source	Register clock : PCLKB AES core clock : PCLKA
AES core function	<ul style="list-style-type: none"> <li>Utilize for on-the-fly decryption of encrypted software stored in external memory.</li> <li>Block size: 128-bit</li> <li>Key size: 128-bit, 192-bit, 256-bit</li> <li>Support the following block cipher mode.                             <ul style="list-style-type: none"> <li>Counter (CTR) mode following NIST SP800-38A</li> </ul> </li> <li>Support side channel counter measure function.</li> <li>Supports self-test function.</li> <li>Counter[127:0] = {IV[127:28], Address[31:4]}, where Address is the memory mapped address of the encrypted data, aligned to the AES block size of 128 bits (16 bytes). The IV shall be chosen according to the recommendations provided in appendix B of SP800-38A. See section 35, Octal Serial Peripheral Interface (OSPI) and section 4.1. Address Space for details. See NIST SP800-38A for details of AES counter mode operation.</li> </ul>
Tamper Resistance	Countermeasures available for side-channel attacks, including SPA/DPA and timing attacks
Module-stop function	Module-stop state can be set to reduce power consumption. same as OSPI module stop
TrustZone Filter	Security and Privilege attribution can be set for each channel. same as TZF of OSPI

### 36.2 Block Diagram

The block diagram of DOTF is shown in Figure 36.1. See Figure 35.1 for a diagram including the OSPI.



**Figure 36.1 Block diagram**

### 36.3 Register Descriptions

DOTF allows the entire area to be an accessible area, but the area where decryption processing is performed can be specified as the entire area or one contiguous area. The region can be specified by setting the start and end addresses to the CONVAREAST and CONAREAED registers, respectively.

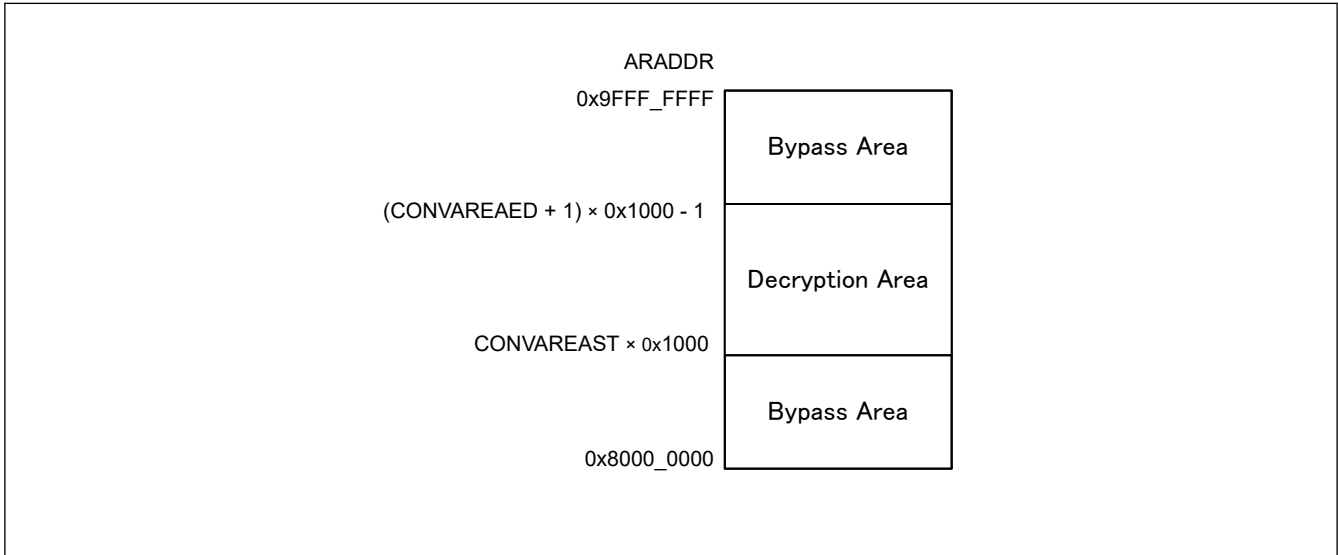


Figure 36.2 Image of decryption area setting

#### 36.3.1 CONVAREAST : DOTF Conversion Area Start Address Register

Base address: DOTF0 = 0x4026\_8800  
 DOTF0\_NS = 0x5026\_8800

Offset address: 0x00



Value after reset: 0

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 0.	R/W
31:12	CONVAREAST[31:12]	The first address of the decryption processing area. The actual address is CONVAREAST[31:12] × 0x1000.	R/W

Note: S-TYPE-3, P-TYPE-3

Specify the first address of the decryption processing area. Set before the AXI transfer request and do not make any further changes. Setting CONVAREAST[31:12] > CONVAREAED[31:12] is prohibited.

#### 36.3.2 CONVAREAED : DOTF Conversion Area End Address Register

Base address: DOTF0 = 0x4026\_8800  
 DOTF0\_NS = 0x5026\_8800

Offset address: 0x04



Value after reset: 0

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 0.	R/W
31:12	CONVAREAED[31:12]	The end address of the decryption processing area. The actual address is CONVAREAED[31:12] × 0x1000.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Setting CONVAREAST[31:12] > CONVAREAED[31:12] is prohibited.

Specify the tail address of the decryption processing area. Set before the AXI transfer request and do not make any further changes. Setting CONVAREAST[31:12] > CONVAREAED[31:12] is prohibited.

### 36.4 Operation

DOTF has one AXI Read Slave Interface and one AXI Read Master Interface.

An AXI transfer request received on the AXI Slave Address channel is issued to the AXI Master's Read Address channel, and the Slave data is received from the Master's AXI Read data channel and sent to the AXI Slave Read data channel.

At this time, the necessity of decryption is determined by the requested address area, and the decrypted data is transmitted through AES core if necessary. When decryption is not required, the received response is sent without conversion.

### 36.5 System flow

The initialization flow after HW reset is shown in [Figure 36.3](#).

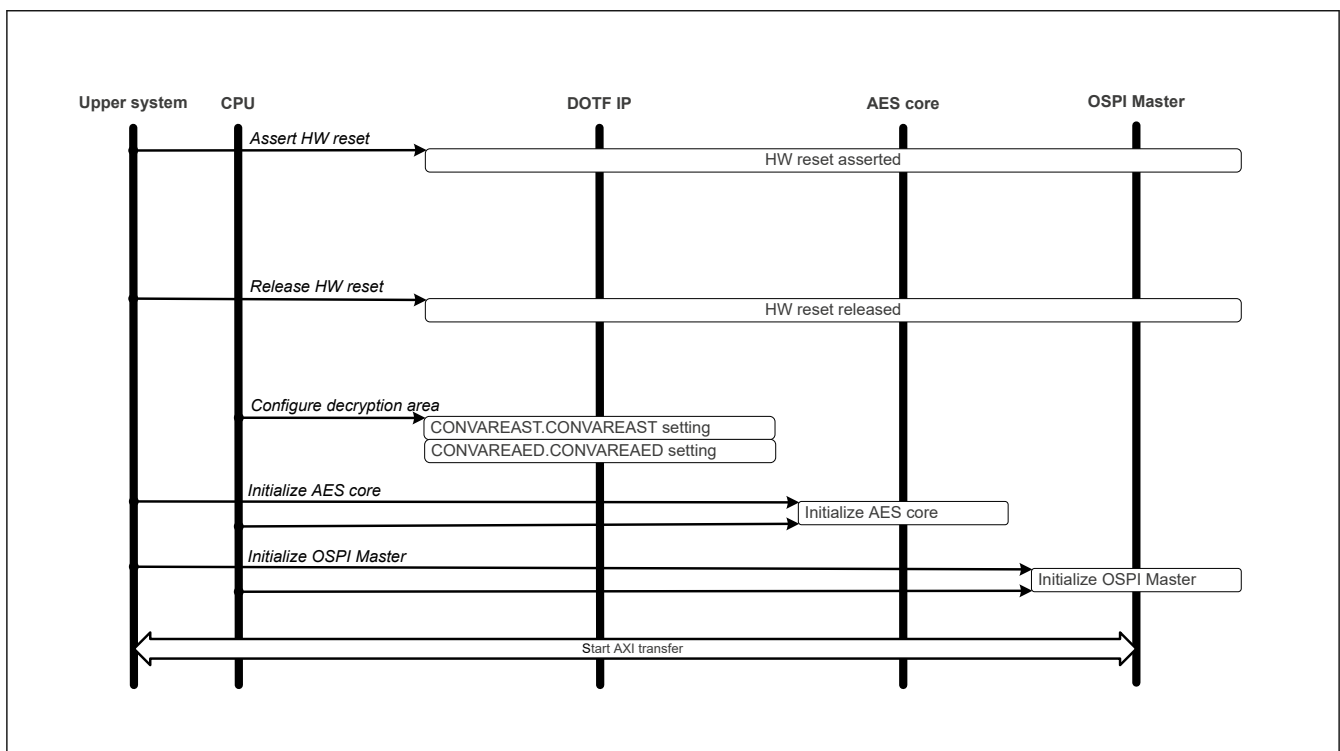


Figure 36.3 Initialization flow

### 36.6 Usage notes

#### 36.6.1 Module-stop function

DOTF operation can be disabled or enabled using Module Stop Control Register B (MSTPCRB). The DOTF module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).



## 37. Serial Sound Interface Enhanced (SSIE)

### 37.1 Overview

The Serial Sound Interface Enhanced (SSIE) can transmit and receive audio data to and from various devices that support any of audio data formats, such as I<sup>2</sup>S, monaural, and TDM.

Table 37.1 lists the SSIE specifications, Figure 37.3, Figure 37.4 and Figure 37.5 show a block diagram of SSIE, and Table 37.4 lists the I/O pins.

**Table 37.1 SSIE specifications**

Item		Description
Number of channels		Two channels, SSIE0 and SSIE1
Communication mode		<ul style="list-style-type: none"> <li>Master/slave</li> <li>Transmission/reception (SSIE0:full duplex communication, SSIE1:half-duplex communication)</li> </ul>
Communication format		<ul style="list-style-type: none"> <li>I<sup>2</sup>S format</li> <li>Monaural format</li> <li>TDM format</li> </ul>
Serial data		<ul style="list-style-type: none"> <li>MSB first</li> <li>Data can be left-justified or right-justified.</li> <li>Data delay (1 clock cycle) or no delay selectable for the period from SSILRCKn/SSIFSn (n = 0, 1) to SSITXD0/SSIRXD0/SSIDATA1</li> <li>System word length: 8, 16, 24, 32, 48, 64, 128, or 256 bits</li> <li>Data word length: 8, 16, 18, 20, 22, 24, or 32 bits</li> <li>Padding polarity: Low or high</li> </ul>
Bit clock (SSIBCKn (n = 0, 1))	In master mode	<ul style="list-style-type: none"> <li>Two clock sources available (AUDIO_CLK/GPT output (GTIOC2A))</li> <li>Clock source division ratio: 1/1, 1/2, 1/4, 1/6, 1/8, 1/12, 1/16, 1/24, 1/32, 1/48, 1/64, 1/96, and 1/128.</li> <li>Supply/stop is selectable while communication is halted.</li> </ul>
	In master/slave mode	<ul style="list-style-type: none"> <li>Polarity (rising edge or falling edge) selectable</li> </ul>
LR clock/frame synchronization (SSILRCKn/SSIFSn (n = 0, 1))	In master mode	<ul style="list-style-type: none"> <li>Polarity (low level or high level) selectable</li> <li>Supply/stop is selectable while communication is halted.</li> </ul>
Transmit data (SSITXD0/SSIDATA1) and receive data (SSIRXD0/SSIDATA1)	Transmission	<ul style="list-style-type: none"> <li>Muting method (transmission of transmit FIFO data or transmission of data fixed to 0) selectable</li> </ul>
FIFO	Capacity	<ul style="list-style-type: none"> <li>Transmit FIFO/receive FIFO: 4 bytes × 32 stages</li> </ul>
	Data alignment	<ul style="list-style-type: none"> <li>Data alignment method (left-justification or right-justification) selectable for the data transfer between FIFO and shift register</li> </ul>
Interrupt	Interrupt output	<ul style="list-style-type: none"> <li>Communication error/idle mode</li> <li>Receive data full</li> <li>Transmit data empty</li> </ul>
Low power consumption function		<ul style="list-style-type: none"> <li>Whether to supply the audio clock selectable in master mode</li> </ul>
Module stop function		Module stop state can be set for each channels to reduce power consumption.
TrustZone Filter		Security and Privilege attribution can be set for each channels.

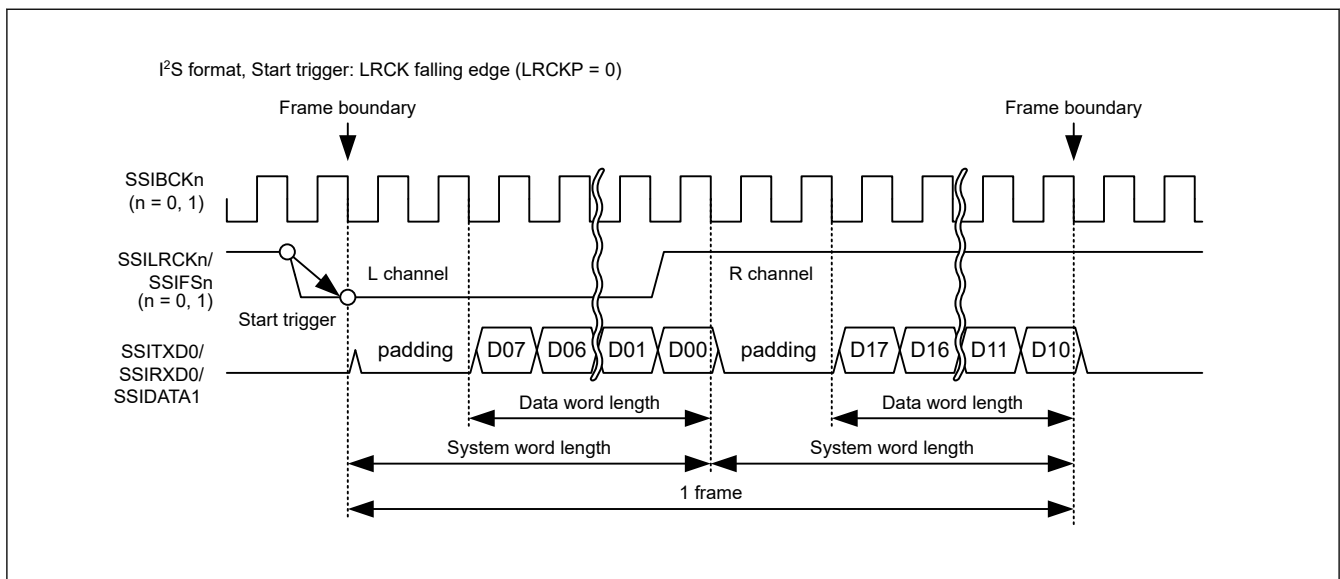
The following table lists and defines the terms used for the communication formats SSIE can use:

**Table 37.2 Definition of terms (1 of 2)**

Term	Definition
Start trigger	First edge of the signal on the SSILRCKn/SSIFSn (n = 0, 1) pin when the signal is set to the value specified in LRCKP to enable communication

**Table 37.2 Definition of terms (2 of 2)**

Term	Definition
Frame boundary	Point where SSIE starts transferring the first data of a frame or the point where SSIE ends transferring the last data of the frame
Frame word number	Number of sound channels per frame
System word length	Number of bits per channel
Data word length	Number of significant bits per channel
Control bits for communication formats	<ul style="list-style-type: none"> <li>• SSICR register: FRM, DWL, SWL, LRCKP, SPDP, SDTA, PDTA, and DEL bits</li> <li>• SSIFCR register: BSW bit</li> <li>• SSIOFR register: OMOD bit</li> <li>• SSISCR register: TDES[4:0] and RDFS[4:0] bits</li> </ul>



**Figure 37.1 Definition of communication format**

Figure 37.2 and Figure 37.3 shows a block diagram of SSIE.

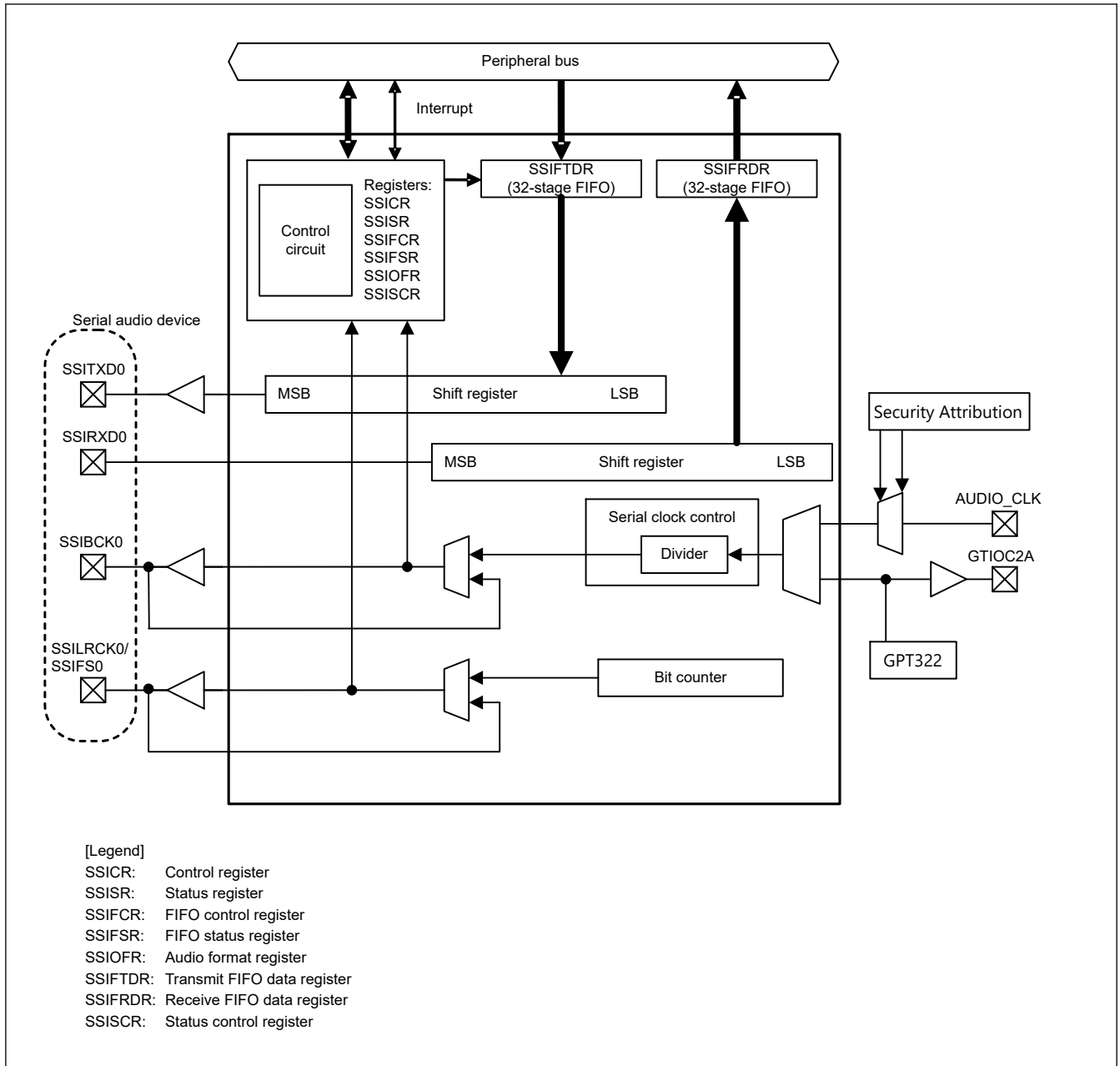
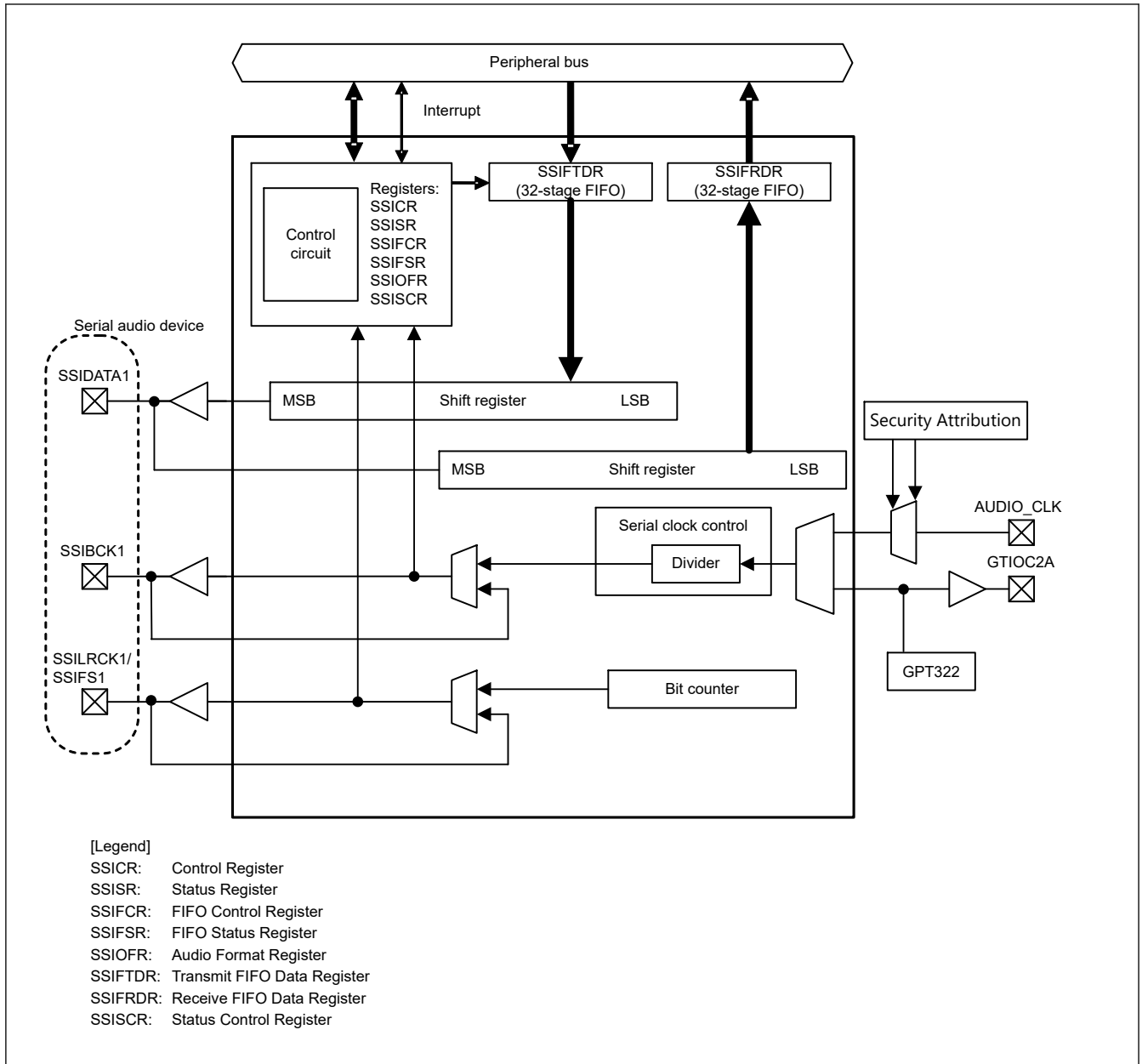
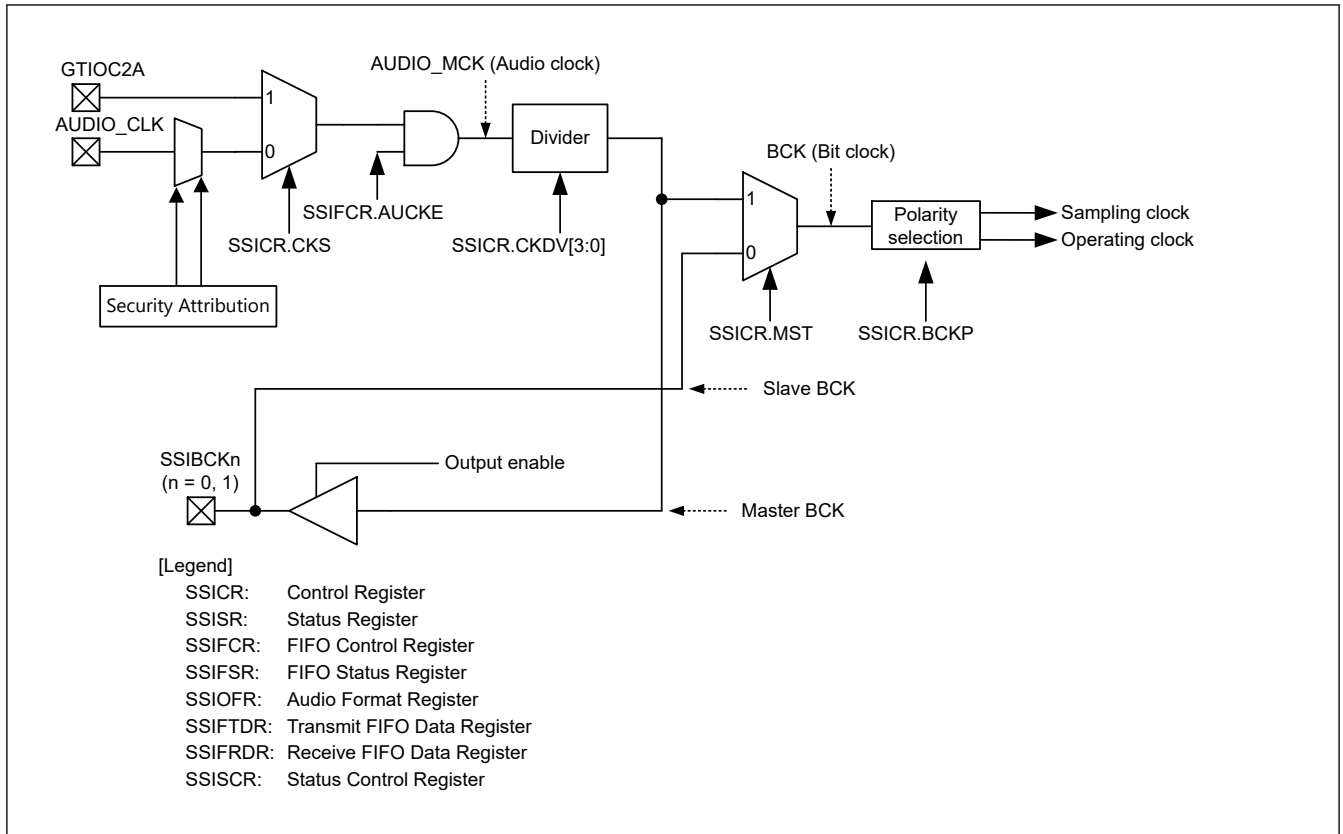


Figure 37.2 SSIE block diagram (SSIE0)



**Figure 37.3 SSIE block diagram (SSIE1)**

Figure 37.4 shows the clock configuration of SSIE.



**Figure 37.4 SSIE clock configuration**

The AUDIO\_CLK can be input according to the security attribution of Port and SSIE and SSICR.CKS bit setting as shown in the [Table 37.3](#).

**Table 37.3 The input condition of AUDIO\_CLK by the security setting**

PORT Security Attribution (PmSAR (m = 0 - 9, A, B))	SSIE0 Security Attribution (PSARm (B - E))	SSIE0's SSICR.CKS	SSIE1 Security Attribution (PSARm (B - E))	SSIE1's SSICR.CKS	AUDIO_CLK pin input
0 (secure)	0 (secure)	0 (AUDIO_CLK)	Don't care		AUDIO_CLK input enable
	Don't care		0 (secure)	0 (AUDIO_CLK)	
1 (non-secure)	1 (non-secure)	Don't care	0 (secure)	1 (GTIOC2A)	AUDIO_CLK input disable
		Don't care	1 (non-secure)	Don't care	
	0 (secure)	1 (GTIOC2A)	1 (non-secure)	Don't care	
	1 (non-secure)	Don't care			
other than above					AUDIO_CLK input disable

[Table 37.4](#) lists the I/O pins.

**Table 37.4 SSIE I/O pins**

Channel	Pin name	I/O	Function
Common	SSIBLn	Input	Serial bit clock pins
	AUDIO_CLK	Input	External clock pin for audio (input oversampling clock)
	SSILRCKn/SSIFSn	Output	LR clock/frame synchronization pins
SSIE0	SSIRXD0	Input	Serial data input pin
	SSITXD0	Output	Serial data output pin
SSIE1	SSIDATA1	I/O	Serial data input/output pin

Note: n = 0, 1

## 37.2 Register Descriptions

### 37.2.1 SSICR : Control Register

Base address: SSIE<sub>n</sub> = 0x4025\_D000 + 0x0100 × n (n = 0, 1)  
 SSIE<sub>n</sub>\_NS = 0x5025\_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CKS	TUIEN	TOIEN	RUIEN	ROIEN	IEN	—	FRM[1:0]	DWL[2:0]			SWL[2:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MST	BCKP	LRCK <sub>P</sub>	SPDP	SDTA	PDTA	DEL	CKDV[3:0]				MUEN	—	TEN	REN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	REN	Reception Enable*2 0: Disables reception 1: Enables reception (starts reception)	R/W
1	TEN	Transmission Enable*2 0: Disables transmission 1: Enables transmission (starts transmission)	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	MUEN	Mute Enable 0: Disables muting on the next frame boundary 1: Enables muting on the next frame boundary	R/W
7:4	CKDV[3:0]	Selects Bit Clock Division Ratio*1 0x0: AUDIO_MCK 0x1: AUDIO_MCK/2 0x2: AUDIO_MCK/4 0x3: AUDIO_MCK/8 0x4: AUDIO_MCK/16 0x5: AUDIO_MCK/32 0x6: AUDIO_MCK/64 0x7: AUDIO_MCK/128 0x8: AUDIO_MCK/6 0x9: AUDIO_MCK/12 0xA: AUDIO_MCK/24 0xB: AUDIO_MCK/48 0xC: AUDIO_MCK/96 Others: Setting prohibited	R/W
8	DEL	Selects Serial Data Delay*1 In the monaural format, this bit controls the waveform of SSILRCKn/SSIFS <sub>n</sub> (n = 0, 1). For details, see <a href="#">section 37.3.2. Monaural Format</a> . 0: Delay of 1 cycle of SSIBCK <sub>n</sub> (n = 0, 1) between SSILRCKn/SSIFS <sub>n</sub> (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1 1: No delay between SSILRCKn/SSIFS <sub>n</sub> (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1	R/W
9	PDTA	Selects Placement Data Alignment*1 0: Left-justifies placement data (SSIFTDR, SSIFRDR) 1: Right-justifies placement data (SSIFTDR, SSIFRDR)	R/W
10	SDTA	Selects Serial Data Alignment*1 0: Transmits and receives serial data first and then padding bits 1: Transmit and receives padding bits first and then serial data	R/W

Bit	Symbol	Function	R/W																				
11	SPDP	Selects Serial Padding Polarity <sup>*1</sup> 0: Padding data is at a low level 1: Padding data is at a high level	R/W																				
12	LRCKP	Selects the Initial Value and Polarity of LR Clock/Frame Synchronization Signal <sup>*1</sup> 0: The initial value is at a high level. The start trigger for a frame is synchronized with a falling edge of SSILRCKn/SSIFS <sub>n</sub> (n = 0, 1). 1: The initial value is at a low level. The start trigger for a frame is synchronized with a rising edge of SSILRCKn/SSIFS <sub>n</sub> (n = 0, 1).	R/W																				
13	BCKP	Selects Bit Clock Polarity <sup>*1</sup> 0: SSILRCKn/SSIFS <sub>n</sub> (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1 change at a falling edge (SSILRCKn/SSIFS <sub>n</sub> (n = 0, 1) and SSIRXD0/SSIDATA1 are sampled at a rising edge of SSIBCKn (n = 0, 1)). 1: SSILRCKn/SSIFS <sub>n</sub> (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1 change at a rising edge (SSILRCKn/SSIFS <sub>n</sub> (n = 0, 1) and SSIRXD0/SSIDATA1 are sampled at a falling edge of SSIBCKn (n = 0, 1)).	R/W																				
14	MST	Master Enable <sup>*1</sup> 0: Slave-mode communication 1: Master-mode communication	R/W																				
15	—	This bit is read as 0. The write value should be 0.	R/W																				
18:16	SWL[2:0]	Selects System Word Length <sup>*1</sup> 0 0 0: 8 bits 0 0 1: 16 bits 0 1 0: 24 bits 0 1 1: 32 bits 1 0 0: 48 bits 1 0 1: 64 bits 1 1 0: 128 bits 1 1 1: 256 bits	R/W																				
21:19	DWL[2:0]	Selects Data Word Length <sup>*1</sup> 0 0 0: 8 bits 0 0 1: 16 bits 0 1 0: 18 bits 0 1 1: 20 bits 1 0 0: 22 bits 1 0 1: 24 bits 1 1 0: 32 bits 1 1 1: Setting prohibited	R/W																				
23:22	FRM[1:0]	Selects Frame Word Number <sup>*1</sup> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="4">Communication format (SSIOFR.OMOD[1:0])</th> </tr> <tr> <th>FRM[1:0]</th> <th>I<sup>2</sup>S (00b)</th> <th>Monaural (10b)</th> <th>TDM (01b)</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>2</td> <td>1</td> <td>Setting prohibited</td> </tr> <tr> <td>01b</td> <td rowspan="3">Setting prohibited</td> <td rowspan="3">Setting prohibited</td> <td>4</td> </tr> <tr> <td>10b</td> <td>6</td> </tr> <tr> <td>11b</td> <td>8</td> </tr> </tbody> </table>	Communication format (SSIOFR.OMOD[1:0])				FRM[1:0]	I <sup>2</sup> S (00b)	Monaural (10b)	TDM (01b)	00b	2	1	Setting prohibited	01b	Setting prohibited	Setting prohibited	4	10b	6	11b	8	R/W
Communication format (SSIOFR.OMOD[1:0])																							
FRM[1:0]	I <sup>2</sup> S (00b)	Monaural (10b)	TDM (01b)																				
00b	2	1	Setting prohibited																				
01b	Setting prohibited	Setting prohibited	4																				
10b			6																				
11b			8																				
24	—	This bit is read as 0. The write value should be 0.	R/W																				
25	IIEN	Idle Mode Interrupt Output Enable 0: Disables idle mode interrupt output 1: Enables idle mode interrupt output	R/W																				
26	ROIEN	Receive Overflow Interrupt Output Enable 0: Disables receive overflow interrupt output 1: Enables receive overflow interrupt output	R/W																				

Bit	Symbol	Function	R/W
27	RUIEN	Receive Underflow Interrupt Output Enable 0: Disables receive underflow interrupt output 1: Enables receive underflow interrupt output	R/W
28	TOIEN	Transmit Overflow Interrupt Output Enable 0: Disables transmit overflow interrupt output 1: Enables transmit overflow interrupt output	R/W
29	TUIEN	Transmit Underflow Interrupt Output Enable 0: Disables transmit underflow interrupt output 1: Enables transmit underflow interrupt output	R/W
30	CKS	Selects an Audio Clock for Master-mode Communication*1 0: Selects the AUDIO_CLK input 1: Selects the GTIOC2A (GPT output)	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Writing to these bits while SSIE is in a communication state (SSISR.IIRQ = 0) is prohibited. If the value of these bits is changed by rewriting, subsequent operation is unpredictable.

Note 2. If the TEN bit or REN bit is rewritten, make sure that the SSISR.IIRQ bit is in the desired status. If the value of the TEN or REN bit is changed by rewriting, subsequent operation is unpredictable. For example, when transmission or reception is enabled, check that SSISR.IIRQ is 0; when transmission or reception is disabled, check that SSISR.IIRQ is 1.

With this register, select an audio clock, control interrupt requests, select data formats, and set an operation mode.

#### TEN and REN bits (Transmission and Reception Enable)

The TEN and REN bits enable/disable transmission and reception. When 1 is written to one of these bits, the corresponding communication operation starts in synchronization with a start trigger by the SSILRCK<sub>n</sub>/SSIFS<sub>n</sub> (n = 0, 1) signal. For details, see [section 37.6.2. Transmission](#) to [section 37.6.4. Transmission and Reception](#). When 0 is written to this bit, the current communication operation stops at the next frame boundary. To use SSIE for both transmission and reception, always write 1 to these bits together. When stopping the communication using SSIE, always disable both transmission and reception (write 0 to the TEN and REN bits).

If you want to stop SSIE before a frame boundary is reached, perform a software reset procedure.

#### MUEN bit (Mute Enable)

The MUEN bit sets/clears the mute function for the data output from the SSITXD0/SSIDATA1 pin. When this bit is set to 1 in the middle of a frame, the SSITXD0/SSIDATA1 output changes to 0 at the next frame boundary. When this bit is set to 0 in the middle of a frame, the SSITXD0/SSIDATA1 output changes to the data of transmit FIFO data register at the next frame boundary. Note that this bit controls data only. Status flags and interrupt signals are normally generated.

Changing the value of this bit must be performed only after setting the communication format to be used.



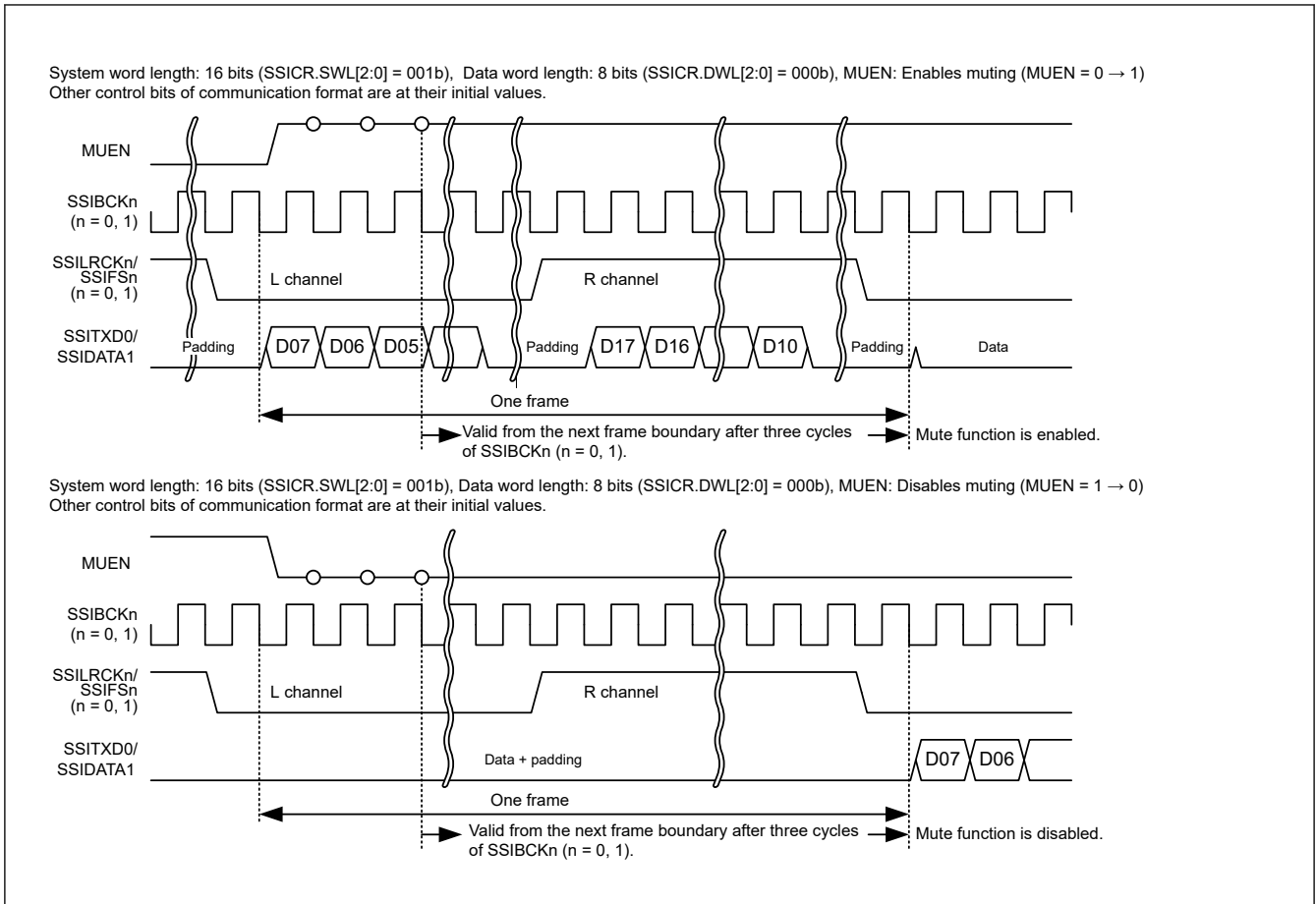


Figure 37.5 Transmit data with the mute function set

**CKDV[3:0] bits (Selects Bit Clock Division Ratio)**

The CKDV[3:0] bits set the division ratio of the bit clock based on AUDIO\_MCK in master-mode communication (MST=1). In slave-mode communication (MST = 0), setting of these bits are invalid.

Writing to this bit must be performed when the supply of AUDIO\_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in [section 37.2.3. SSIFCR : FIFO Control Register](#).

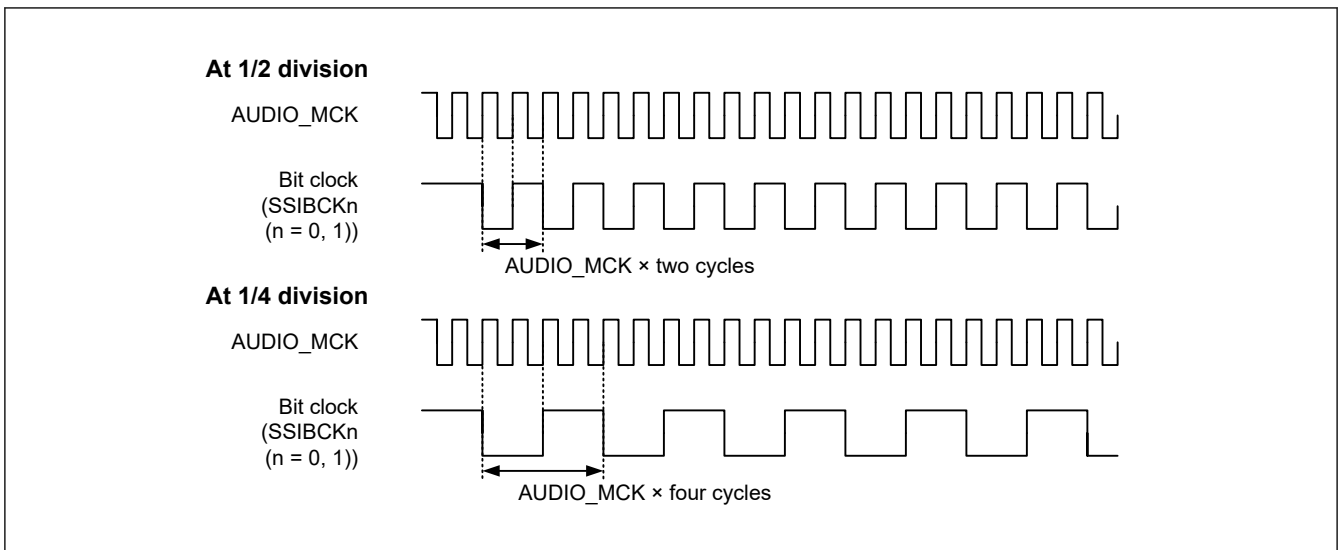
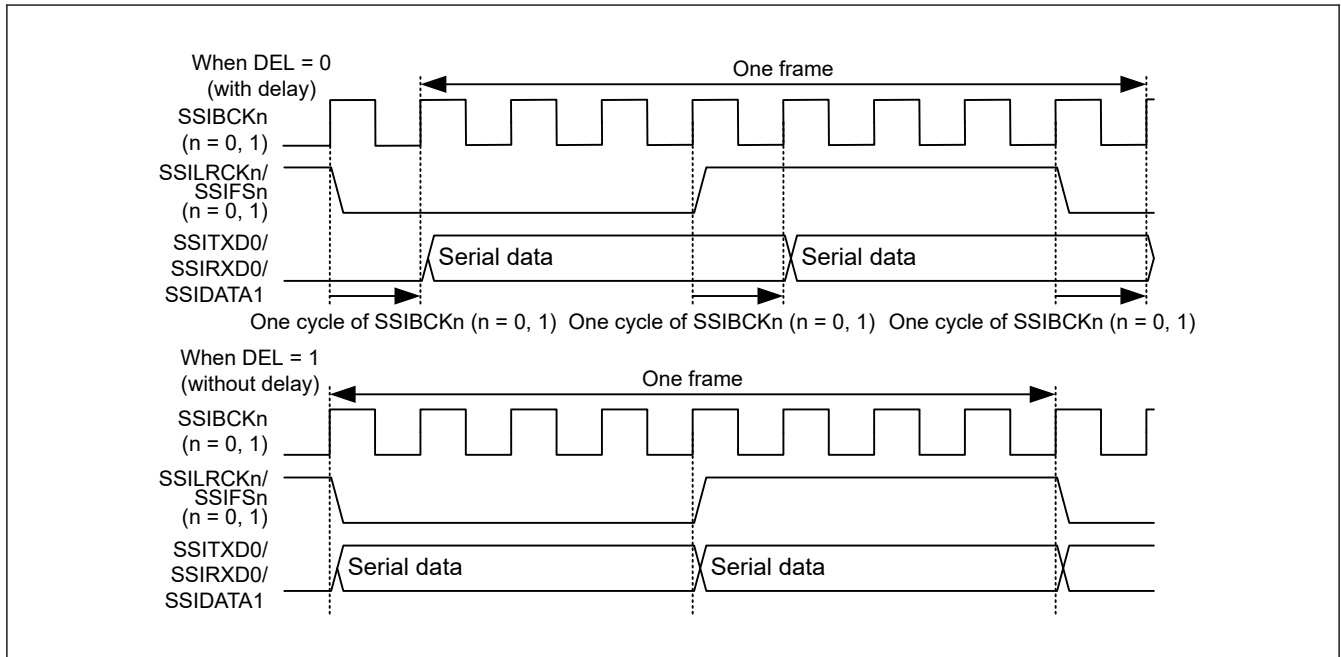


Figure 37.6 Sampling frequencies in master-mode communication

**DEL bit (Selects Serial Data Delay)**

The DEL bit sets whether or not there will be a delay between SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) and SSITXD0/SSIRXD0/SSIDATA1.

For the I<sup>2</sup>S or TDM format, set the DEL bit to 0. When the monaural format is used, setting of this bit changes the high period width of SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1). For details, see [section 37.3.2. Monaural Format](#). When using a compatible communication format, specify a setting of this bit that enables communication.



**Figure 37.7 Setting of delay in serial data**

**PDTA bit (Selects Placement Data Alignment)**

The PDTA bit sets how to align placement data. With the setting of data word length as 32 bits (SSICR.DWL[2:0] = 110b), this bit is invalid.

At transmission, see [Figure 37.8](#).

		First transmission data	Second transmission data	Third transmission data	Fourth transmission data
DWL[2:0]	SSIFTDR				Transmission shift register
		PDTA = 0 (left-justify)		PDTA = 1 (right-justify)	
000 (8 bits)		7	0	Invalid	
		7	0	Invalid	
		7	0	Invalid	
		7	0	Invalid	
001 (16 bits)		15	0	Invalid	
		15	0	Invalid	
		15	0	Invalid	
		15	0	Invalid	
010 to 100 18bit : X = 17 20bit : X = 19 22bit : X = 21 24bit : X = 23		X	0	Invalid	
		X	0	Invalid	
		X	0	Invalid	
		X	0	Invalid	
		X	0	Invalid	
110 (32 bits)		31	0		
		31	0		
		31	0		
		31	0		
111 (Setting prohibited)	Setting prohibited				

Figure 37.8 Alignment of placement data at transmission

At reception, see Figure 37.9.

		First transmission data	Second transmission data	Third transmission data	Fourth transmission data
DWL[2:0]	Receive shift register	SSIFRDR			
		PDTA = 0 (left-justify)		PDTA = 1 (right-justify)	
000 (8 bits)	Invalid	7	0	7	0
	Invalid	7	0	7	0
	Invalid	7	0	7	0
	Invalid	7	0	7	0
001 (16 bits)	Invalid	15	0	15	0
	Invalid	15	0	15	0
	Invalid	15	0	15	0
	Invalid	15	0	15	0
010 to 100 18bit : X = 17 20bit : X = 19 22bit : X = 21 24bit : X = 23	Invalid	X	0	X	0
	Invalid	X	0	X	0
	Invalid	X	0	X	0
	Invalid	X	0	X	0
110 (32 bits)	31	31	0	31	0
	31	31	0	31	0
	31	31	0	31	0
	31	31	0	31	0
111 (Setting prohibited)					

Figure 37.9 Alignment of placement data at reception

**SDTA bit (Selects Serial Data Delay)**

The SDTA bit sets how to align serial data and padding bits. For communication without padding bits, this bit is invalid.

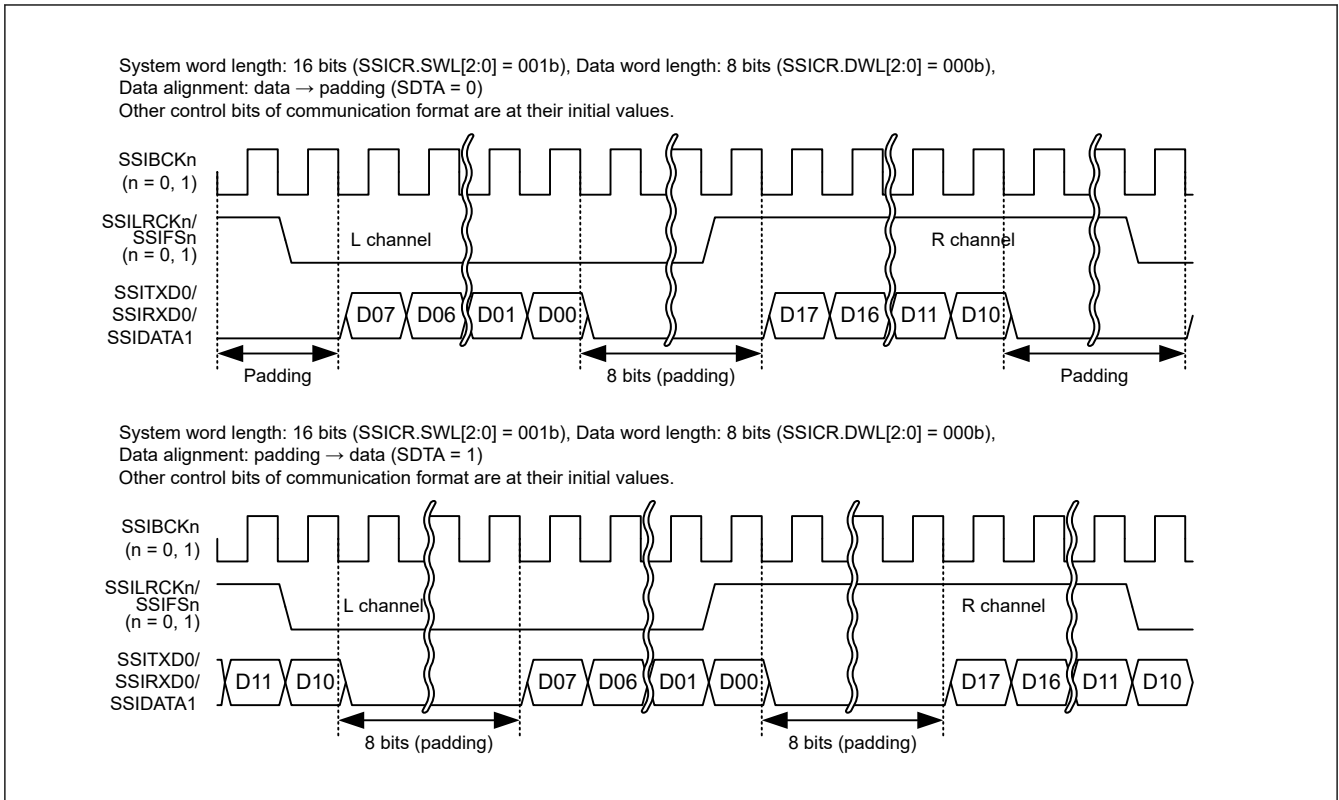


Figure 37.10 Alignment setting of serial data with padding bits

**SPDP bit (Selects Serial Padding Polarity)**

The SPDP bit sets polarity of padding bits.

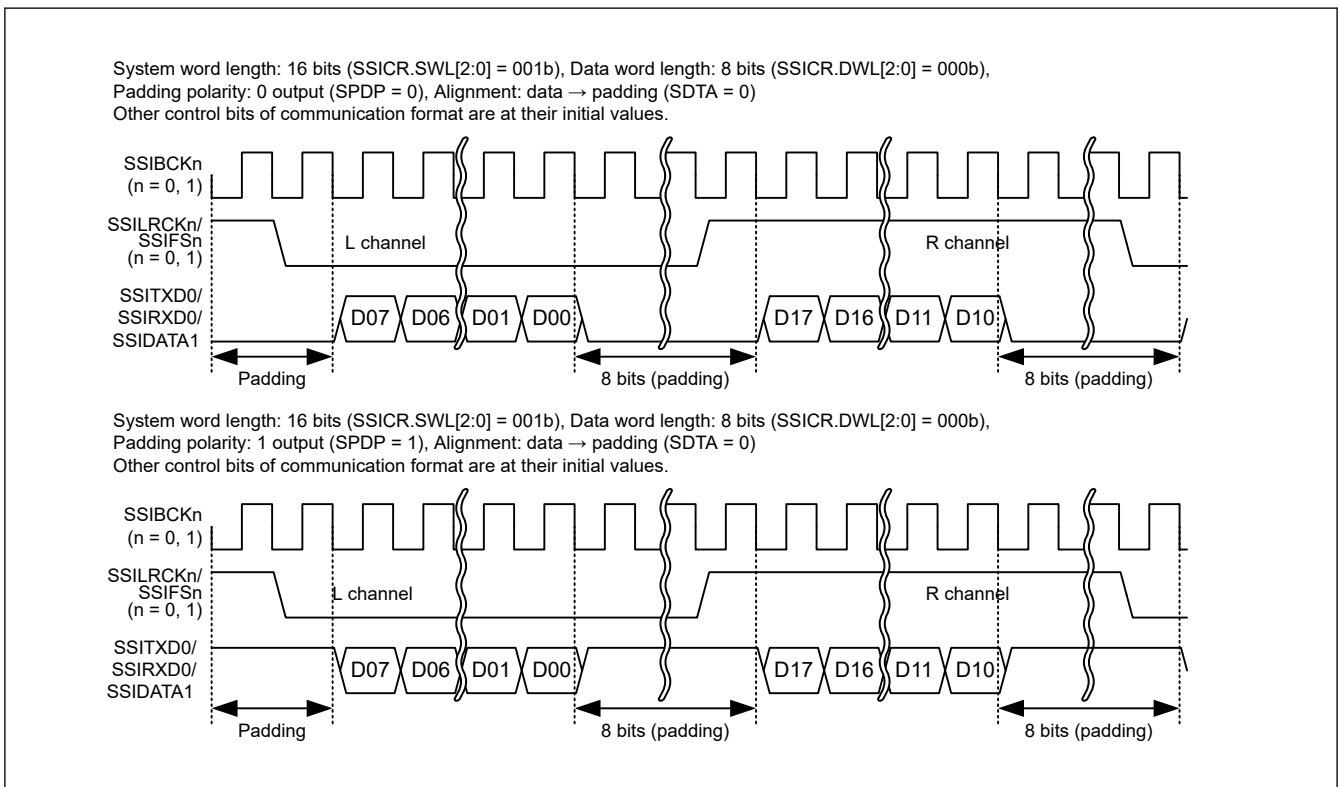


Figure 37.11 Padding bit polarity

**LRCKP bit (Selects the Initial Value and Polarity of LR Clock/Frame Synchronization Signal)**

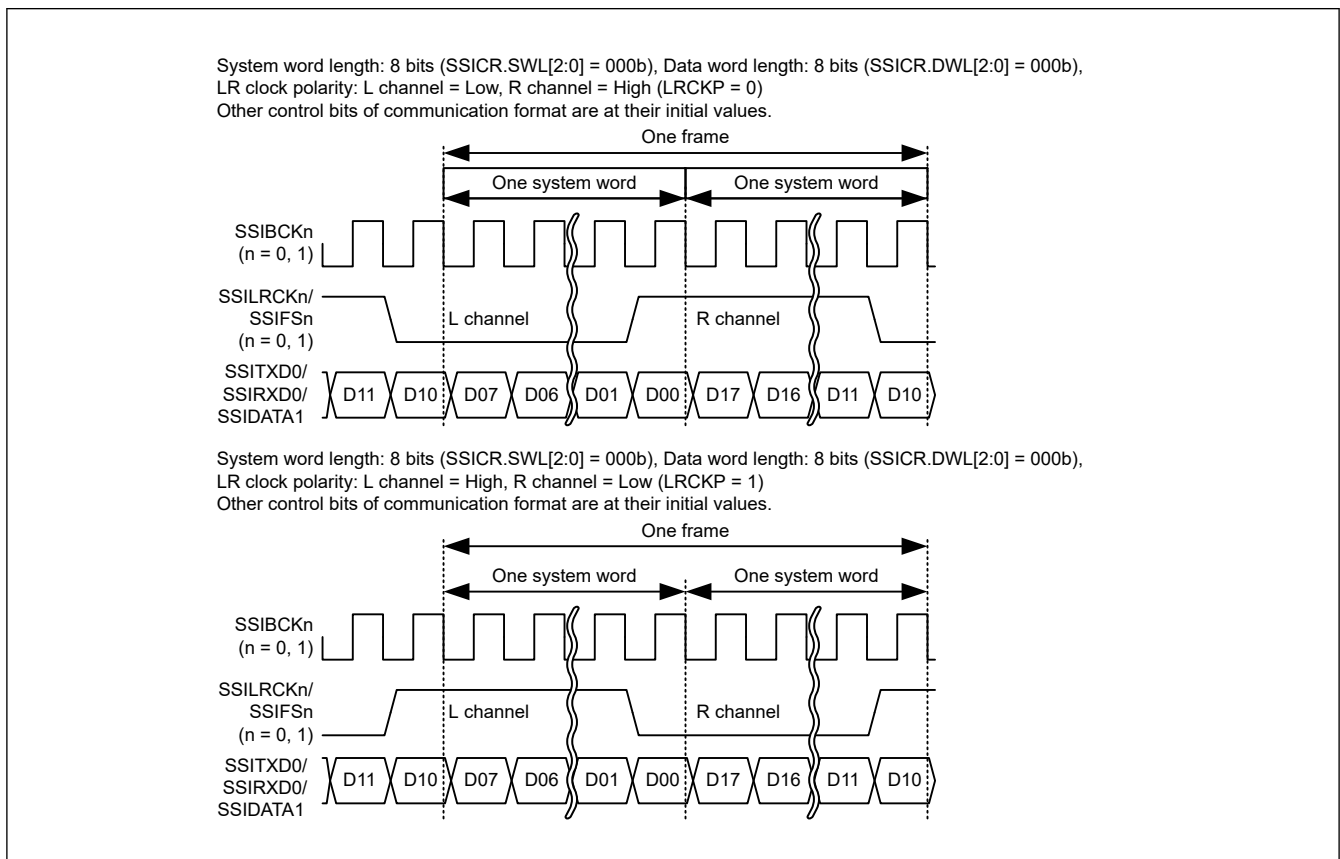
The LRCKP bit sets the initial value and polarity of SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1). Set this bit according to the communication format to be used in SSIE. See [Table 37.5](#) Initial output value and polarity of SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) pin. For the slave-mode communication (MST = 0), only the start trigger is used.

Writing to these bits must be performed when the LR clock supply to the SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in [section 37.2.7. SSIOFR : Audio Format Register](#).

**Table 37.5 Initial output value and polarity of SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) pin**

Communication Format	Expected Initial State	Setting Value of LRCKP
I <sup>2</sup> S	High	0
Monaural	Low	1
TDM	Low	1

Note: When the format to be used is compatible with the I<sup>2</sup>S, monaural, or TDM format, specify settings to enable communication with the respective formats.



**Figure 37.12 LR clock/frame synchronization polarity setting**

**BCKP bit (Selects Bit Clock Polarity)**

The BCKP bit sets the bit clock polarity.

Writing to this bit must be performed when the supply of AUDIO\_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in [section 37.2.3. SSIFCR : FIFO Control Register](#).

**Table 37.6 Bit clock polarity**

Communication	Master/Slave	Timing	BCKP = 0	BCKP = 1
Reception	Slave	At SSILRCKn/SSIFS <sub>n</sub> (n = 0, 1) sampling	SSIBCK <sub>n</sub> (n = 0, 1) rising edge	SSIBCK <sub>n</sub> (n = 0, 1) falling edge
	Master/slave	At SSIRXD0/SSIDATA1 sampling	SSIBCK <sub>n</sub> (n = 0, 1) rising edge	SSIBCK <sub>n</sub> (n = 0, 1) falling edge
Transmission	Master	At change of SSILRCKn/SSIFS <sub>n</sub> (n = 0, 1) output	SSIBCK <sub>n</sub> (n = 0, 1) falling edge	SSIBCK <sub>n</sub> (n = 0, 1) rising edge
	Master/slave	At change of SSITXD0/SSIDATA1 output	SSIBCK <sub>n</sub> (n = 0, 1) falling edge	SSIBCK <sub>n</sub> (n = 0, 1) rising edge

**MST bit (Master Enable)**

The MST bit sets master-/slave-mode communication.

Writing to this bit must be performed when the supply of AUDIO\_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in [section 37.2.3. SSIFCR : FIFO Control Register](#).

**SWL[2:0] bits (Selects System Word Length)**

The SWL[2:0] bits set the number of bits in one system word. Padding bits are sent and received in relation with one data word set with DWL[2:0]. See [Table 37.13](#) for details.

Writing to these bits must be performed when the LR clock supply to the SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in [section 37.2.7. SSIOFR : Audio Format Register](#).

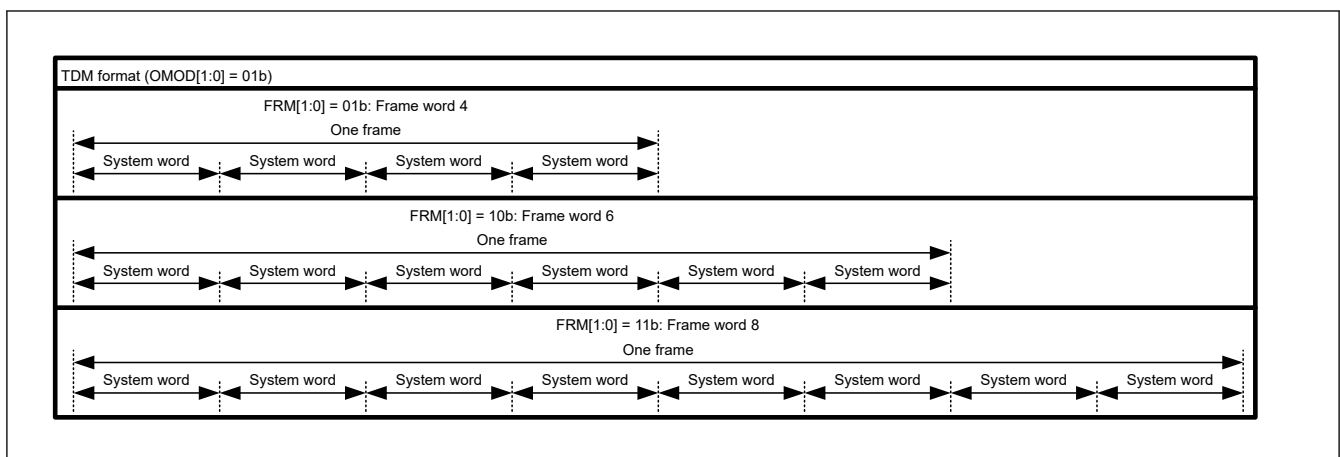
**DWL[2:0] bits (Selects Data Word Length)**

The DWL[2:0] bits set the number of bits in one data word. The data word length (number of bits per data word) must not exceed the system word length (number of bits per system word). For details, see [Table 37.13](#).

**FRM[1:0] bits (Selects Frame Word Number)**

The FRM[1:0] bits set the frame word number in individual communication formats.

Writing to these bits must be performed when the LR clock supply to the SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in [section 37.2.7. SSIOFR : Audio Format Register](#).



**Figure 37.13 Frame word number**

**IEN bit (Idle Mode Interrupt Output Enable)**

The IEN bit enables/disables output of idle mode interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.IIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.IIRQ = 1.

**ROIEN bit (Receive Overflow Interrupt Output Enable)**

The ROIEN bit enables/disables output of receive overflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.ROIEN = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.ROIEN = 1.

**RUIEN bit (Receive Underflow Interrupt Output Enable)**

The RUIEN bit enables/disables output of receive underflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.RUIEN = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.RUIEN = 1.

**TOIEN bit (Transmit Overflow Interrupt Output Enable)**

The TOIEN bit enables/disables output of transmit overflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.TOIEN = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.TOIEN = 1.

**TUIEN bit (Transmit Underflow Interrupt Output Enable)**

The TUIEN bit enables/disables output of transmit underflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.TUIEN = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.TUIEN = 1.

**CKS bit (Selects an Audio Clock for Master-mode Communication)**

The CKS bit sets the audio clock in master-mode communication (MST = 1). In slave-mode communication (MST = 0), setting of this bit is invalid.

Writing to this bit must be performed when the supply of AUDIO\_MCK is stopped. For details about the timing, see the detailed description of the AUCKE bit in [section 37.2.3. SSIFCR : FIFO Control Register](#).

**37.2.2 SSISR : Status Register**

Base address: SSIE<sub>n</sub> = 0x4025\_D000 + 0x0100 × n (n = 0, 1)  
SSIE<sub>n</sub>\_NS = 0x5025\_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	TUIRQ	TOIRQ	RUIRQ	ROIEN	IIRQ	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
24:0	—	These bits are read as 0. The write value should be 0.	R/W
25	IIRQ	Idle Mode Status Flag 0: In the communication state 1: In the idle state	R
26	ROIEN	Receive Overflow Error Status Flag 0: No receive overflow error is generated. 1: A receive overflow error is generated.	R/W
27	RUIEN	Receive Underflow Error Status Flag 0: No receive underflow error is generated. 1: A receive underflow error is generated.	R/W
28	TOIEN	Transmit Overflow Error Status Flag 0: No transmit overflow error is generated. 1: A transmit overflow error is generated.	R/W



Bit	Symbol	Function	R/W
29	TUIRQ	Transmit Underflow Error Status flag 0: No transmit underflow error is generated. 1: A transmit underflow error is generated.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

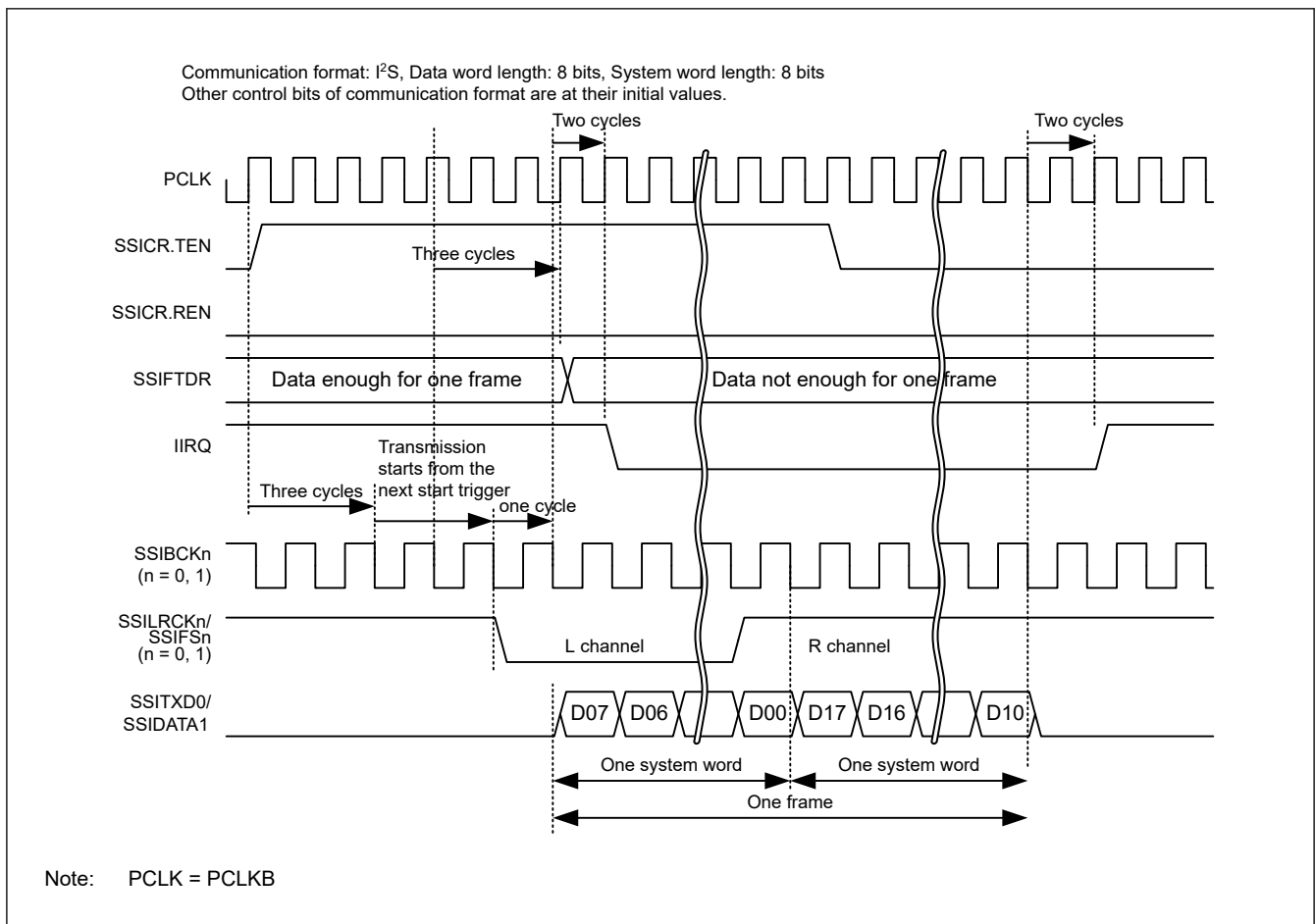
Note: S-TYPE-3, P-TYPE-3

This register is configured with status flags that indicate SSIE operational state.

### IIRQ flag (Idle Mode Status Flag)

The IIRQ flag is a status flag that indicates the idle state. It indicates whether SSIE is in the idle state or communication state.

For details, see [Figure 37.14](#) and [Figure 37.15](#).



**Figure 37.14 IIRQ setting timing (transmission)**

- Transmitter (dedicated to transmission)

#### [Clearing condition]

While transmission was enabled (SSICR.TEN = 1 and SSICR.REN = 0), the transmit data for a transmission frame was written to the SSIFTDR register, and a start trigger was generated by the SSILRCKn/SSIFSs (n = 0, 1) signal.

#### [Clearing timing]

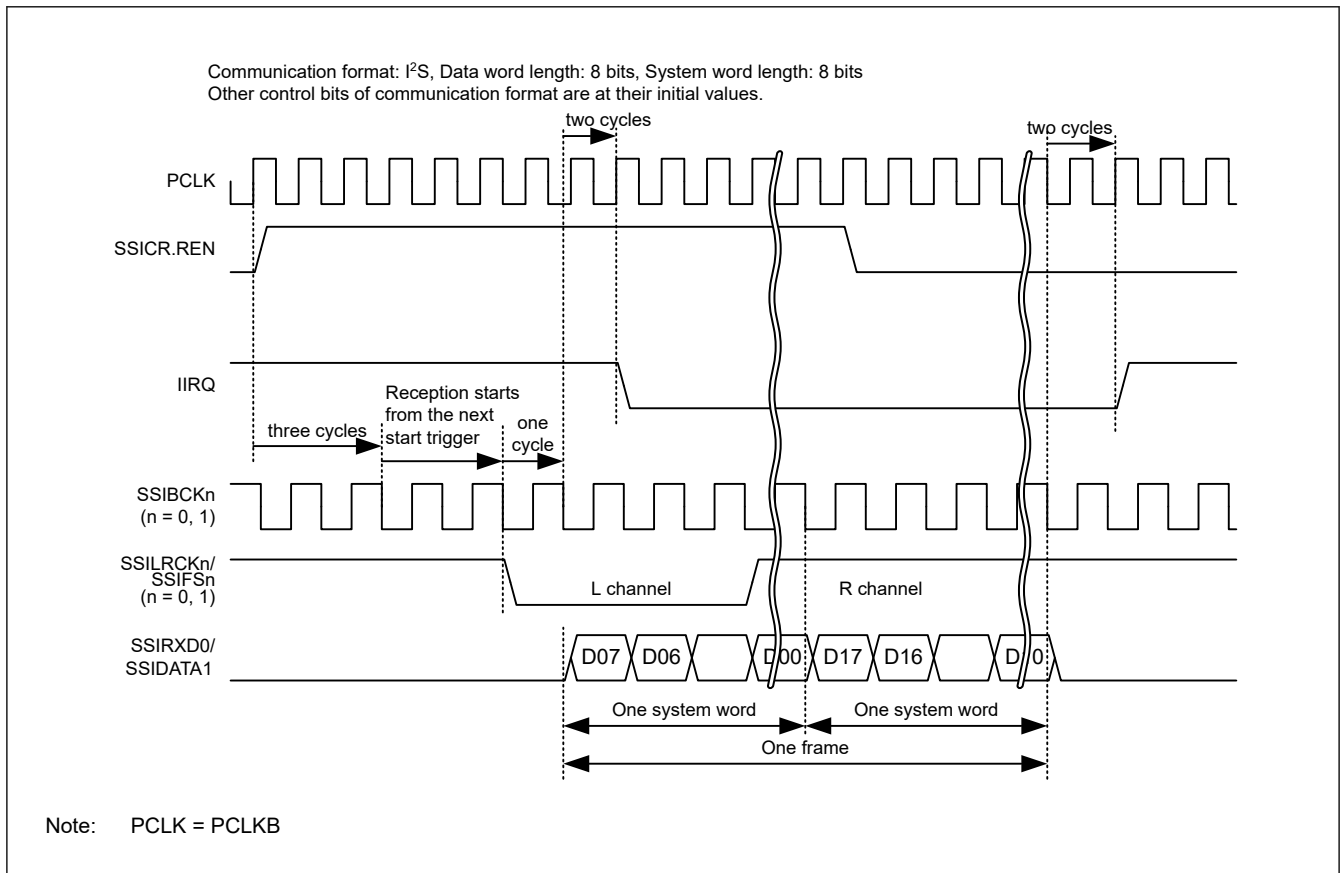
1 SSIBCKn (n = 0, 1) cycle + 2 PCLKB cycles after generation of the start trigger that is the clearing condition.

#### [Setting condition]

While transmission and reception were disabled (SSICR.TEN = 0 and SSICR.REN = 0), transmission of one frame was complete.

[Setting timing]

Two PCLKB cycles after the end of transmission (at a frame boundary) that is the setting condition.



**Figure 37.15 IIRQ setting timing (reception)**

- Receiver (dedicated to reception)

[Clearing condition]

While reception was enabled (SSICR.TEN = 0 and SSICR.REN = 1, a start trigger was generated by the SSILRCKn/SSIFSs (n = 0, 1) signal.

[Clearing timing]

1 SSIBCKn (n = 0, 1) cycle + 2 PCLKB cycles after generation of the start trigger that is the clearing condition.

[Setting condition]

While transmission and reception were disabled (SSICR.TEN = 0 and SSICR.REN = 0), reception of one frame was complete.

[Setting timing]

Two PCLKB cycles after the end of reception (at a frame boundary) that is the setting condition.

- Transceiver (transmission and reception)

[Clearing condition]

While transmission and reception were enabled (SSICR.TEN = 1 and SSICR.REN = 1), the transmit data for a transmission frame was written to the SSIFTDR register, and a start trigger is generated by the SSILRCKn/SSIFSs (n = 0, 1) signal.

[Clearing timing]

1 SSIBCKn (n = 0, 1) cycle + 2 PCLKB cycles after generation of the start trigger that is the clearing condition.

[Setting condition]

While transmission and reception were disabled (SSICR.TEN = 0 and SSICR.REN = 0), transmission of one frame was complete.

[Setting timing]

Two PCLKB cycles after the end of transmission (at a frame boundary) that is the setting condition.

### ROIRQ flag (Receive Overflow Error Status Flag)

The ROIRQ flag is a status flag that indicates a receive overflow error. This flag is set by automatic determination but it must be cleared by register access. This flag indicates that received data is supplied at a higher rate than requested. Data is not transferred from the receive shift register to SSIFRDR where a receive overflow error is generated. For the procedure to recover from the overflow error, see [section 37.6.6. Error Handling](#). This flag is not cleared by a receive FIFO data register reset (SSIFCR.RFRST).

[Priority order for setting and clearing]

Setting is prioritized.\*<sup>1</sup>

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit\*<sup>2</sup>
2. Enabling communication (changing SSICR.REN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition:

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in [Figure 37.19](#))
2. One PCLKB cycle after writing 1 to SSICR.REN.\*<sup>3</sup>

[Setting condition]

At completion of receiving new data while SSIFRDR is full.

[Setting timing]

Three cycles of PCLKB after reception is completed.

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:

- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- One PCLKB cycle passes after 1 has been written to SSICR.REN.

Note 3. After communication is enabled (by changing the value of SSICR.REN bit from 0 to 1), the reception error flags (RUIRQ and ROIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the reception error flags might be unable to be read.

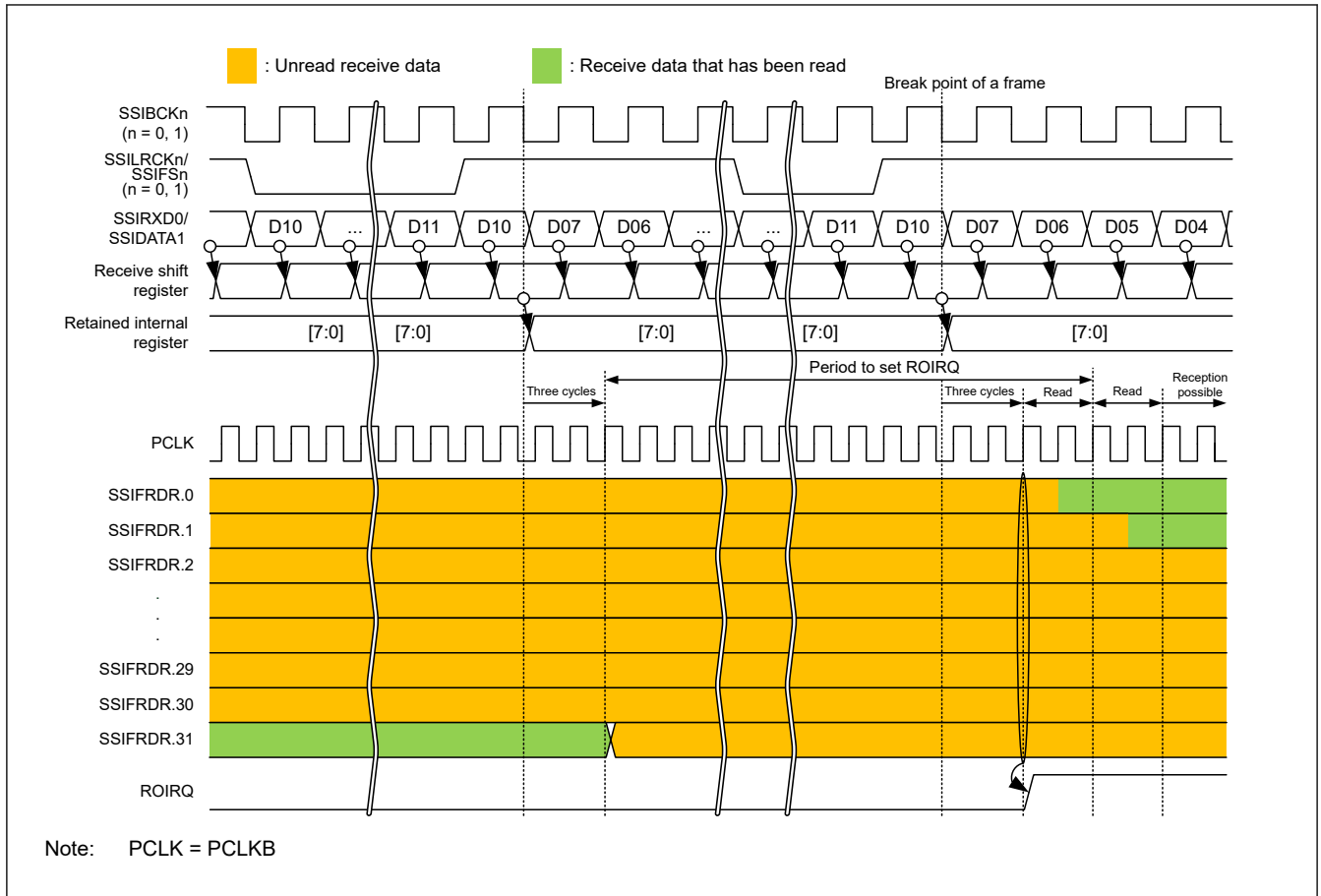


Figure 37.16 ROIRQ setting timing

**RUIRQ flag (Receive Underflow Error Status Flag)**

The RUIRQ flag is a status flag that indicates a receive underflow error. This flag is set by automatic determination but it must be cleared by register access. This flag indicates that SSIFRDR is read while it is empty. Data read from SSIFRDR where a receive underflow error is generated is invalid. See section 37.6.6. Error Handling for the error recovery procedure. This flag is not cleared by a receive FIFO data register reset (SSIFCR.RFRST). Note, however, that this flag is not set even if the SSIFRDR register is read while the receive FIFO data register is reset (by setting SSIFCR.RFRST to 1).

[Priority order for setting and clearing]

Setting is prioritized.\*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit\*2
2. Enabling communication (changing SSICR.REN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 37.19)
2. One PCLKB cycle after writing 1 to SSICR.REN.\*3

[Setting condition]

Reading from SSIFRDR while it is empty.

[Setting timing]

At completion of reading from SSIFRDR. See Figure 37.17.

- Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.
- Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:
- A software reset (SSIFCR.SSIRST = 1) is done.
  - After 1 has been read, writing of 0 is complete.
  - One PCLKB cycle passes after 1 has been written to SSICR.REN.
- Note 3. After communication is enabled (by changing the value of SSICR.REN bit from 0 to 1), the reception error flags (RUIRQ and ROIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the reception error flags might be unable to be read.

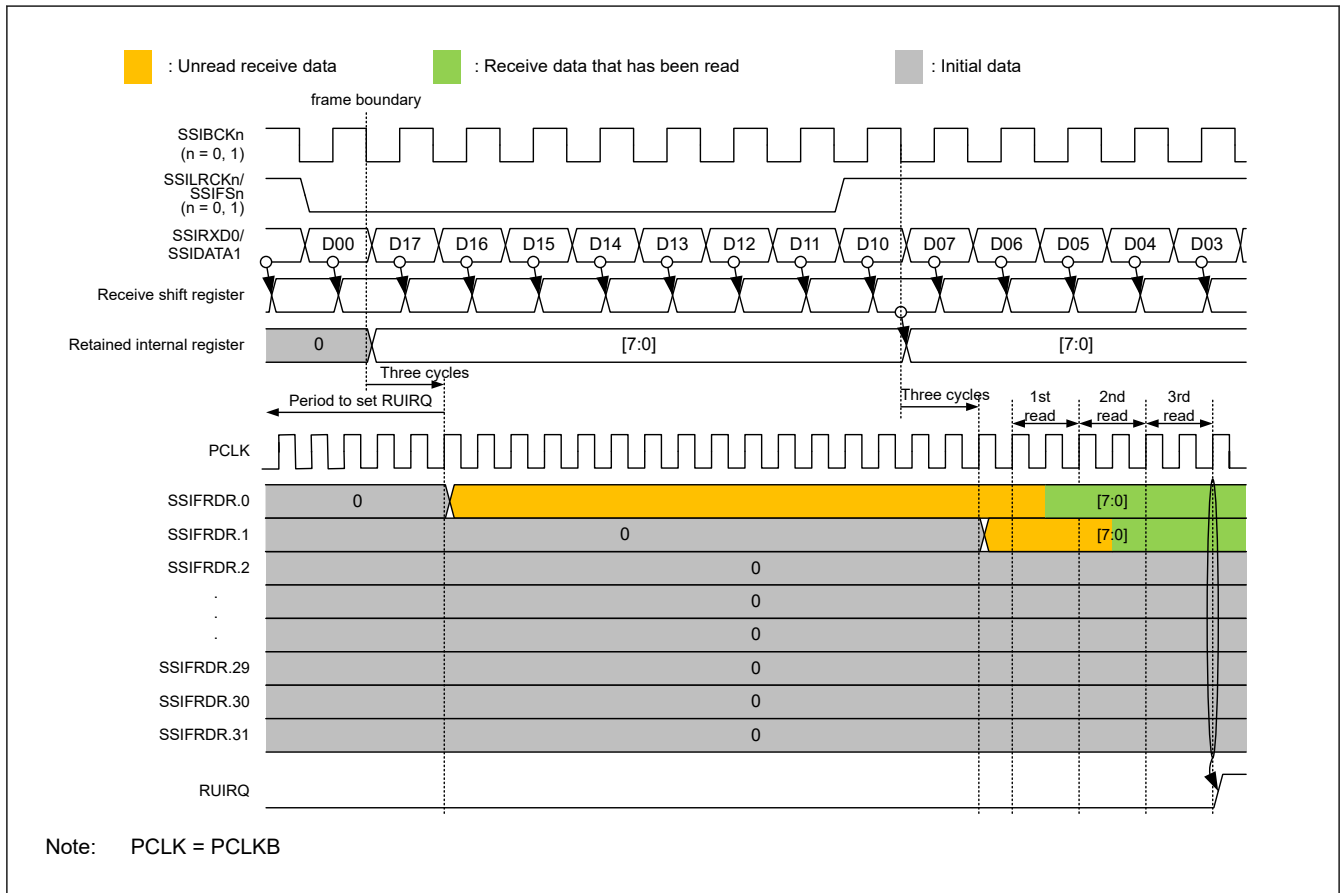


Figure 37.17 RUIRQ setting timing

### TOIRQ flag (Transmit Overflow Error Status Flag)

The TOIRQ flag is a status flag that indicates a transmit overflow error. This flag is set by automatic determination but it must be cleared by register access. This flag indicates that an attempt has been made to write data to the SSIFTDR register when the register is full of data. The data writing that causes a transmit overflow is ignored. For the procedure to recover from the overflow error, see [section 37.6.6. Error Handling](#). This flag is not cleared by a transmit FIFO data register reset (SSIFCR.TFRST).

[Priority order for setting and clearing]

Setting is prioritized.\*1

[Clearing condition]

When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit\*2
2. Enabling communication (changing SSICR.TEN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in [Figure 37.19](#))
2. One PCLKB cycle after writing 1 to SSICR.TEN.\*<sup>3</sup>

[Setting condition]

An attempt was made to write data to the SSIFTDR register when the register is full of data.

[Setting timing]

At completion of writing to SSIFTDR. For details, see [Figure 37.18](#).

- Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.
- Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:
- A software reset (SSIFCR.SSIRST = 1) is done.
  - After 1 has been read, writing of 0 is complete.
  - One PCLKB cycle passes after 1 has been written to SSICR.TEN.
- Note 3. After communication is enabled (by changing the value of SSICR.TEN bit from 0 to 1), the transmission error flags (TOIRQ and TUIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the transmission error flags might be unable to be read.

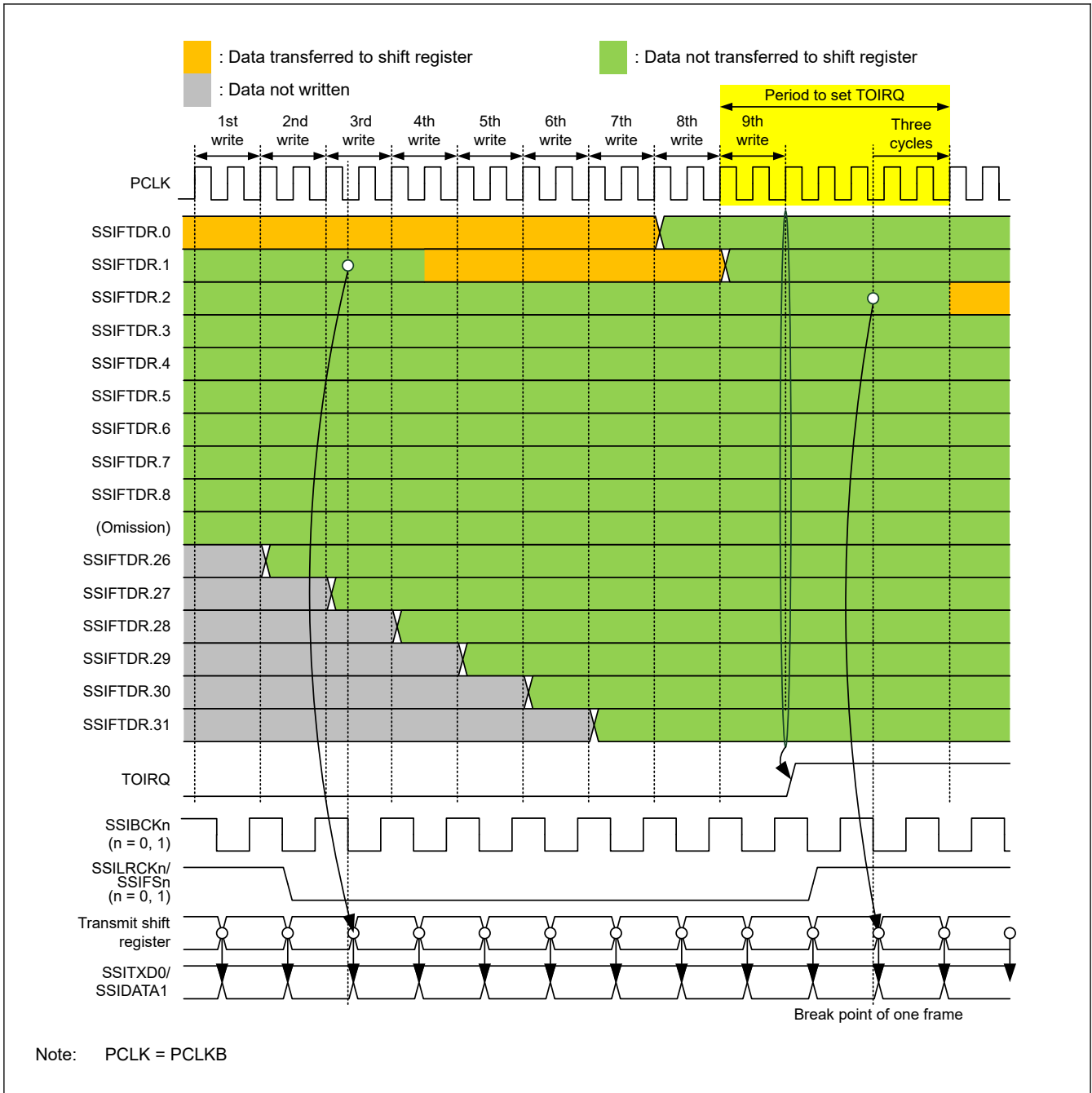


Figure 37.18 TOIRQ setting timing

**TUIRQ flag (Transmit Underflow Error Status flag)**

The TUIRQ flag is a status flag that indicates a transmit underflow error. This flag is set by automatic determination but it must be cleared by register access. This flag indicates that writing the serial data required for a frame to SSIFTDR did not catch up with transmission of the frame. Even if this flag is cleared after it has been set, the SSITXD0/SSIDATA1 output remains to be 0. To output the data written to the transmit FIFO data register (SSIFTDR) to the SSITXD0/SSIDATA1 pin, follow the communication stop procedure in Figure 37.56 and error-handling procedure in Figure 37.57. For the procedure to recover from an error, see section 37.6.6. Error Handling. This flag is not cleared by a reset of transmit FIFO data register (by the SSIFCR.TFRST signal).

[Priority order for setting and clearing]

Setting is prioritized.\*1

[Clearing condition]

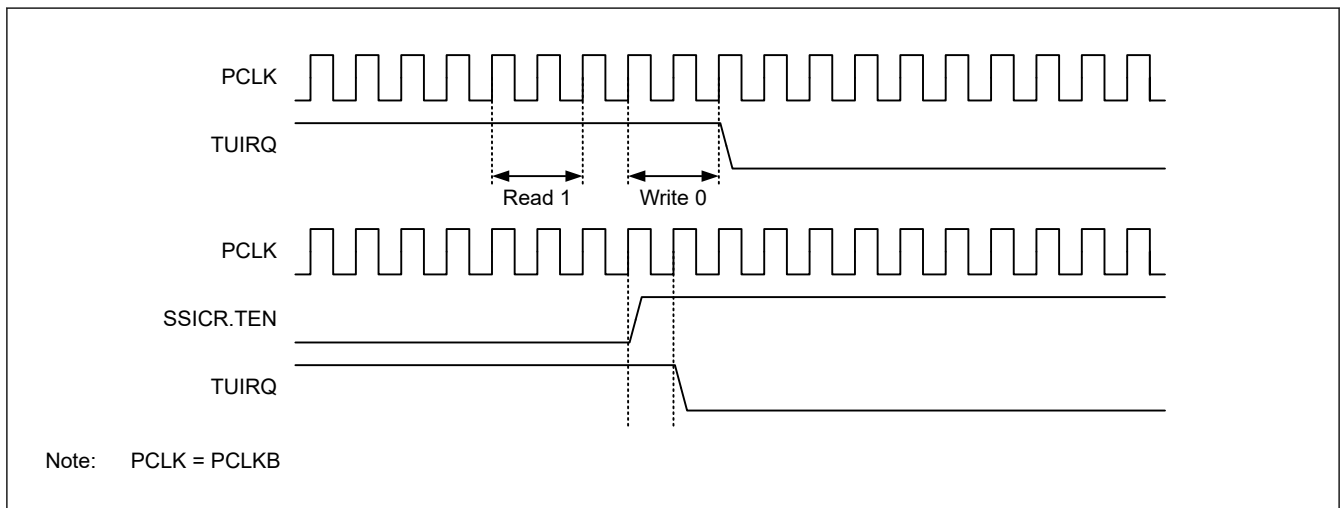
When either of the following operations is done:

1. Writing 0 to this bit after reading 1 from this bit<sup>\*2</sup>
2. Enabling communication (changing SSICR.TEN from 0 to 1).

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit
2. One PCLKB cycle after writing 1 to SSICR.TEN.<sup>\*3</sup>



**Figure 37.19 TUIRQ clearing timing**

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following three conditions is met:

- A software reset (SSIFCR.SSIRST = 1) is done.
- After 1 has been read, writing of 0 is complete.
- One PCLKB cycle passes after 1 has been written to SSICR.TEN.

Note 3. After communication is enabled (by changing the value of SSICR.TEN bit from 0 to 1), the transmission error flags (TOIRQ and TUIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the transmission error flags might be unable to be read.

[Setting condition]

When communication continues over a frame boundary, the transmit data required for the next frame has not been written to SSIFTDR. For details, see [Figure 37.20](#) and [Figure 37.21](#).

[Setting timing]

Three PCLKB cycles after the frame boundary. For details, see [Figure 37.20](#).



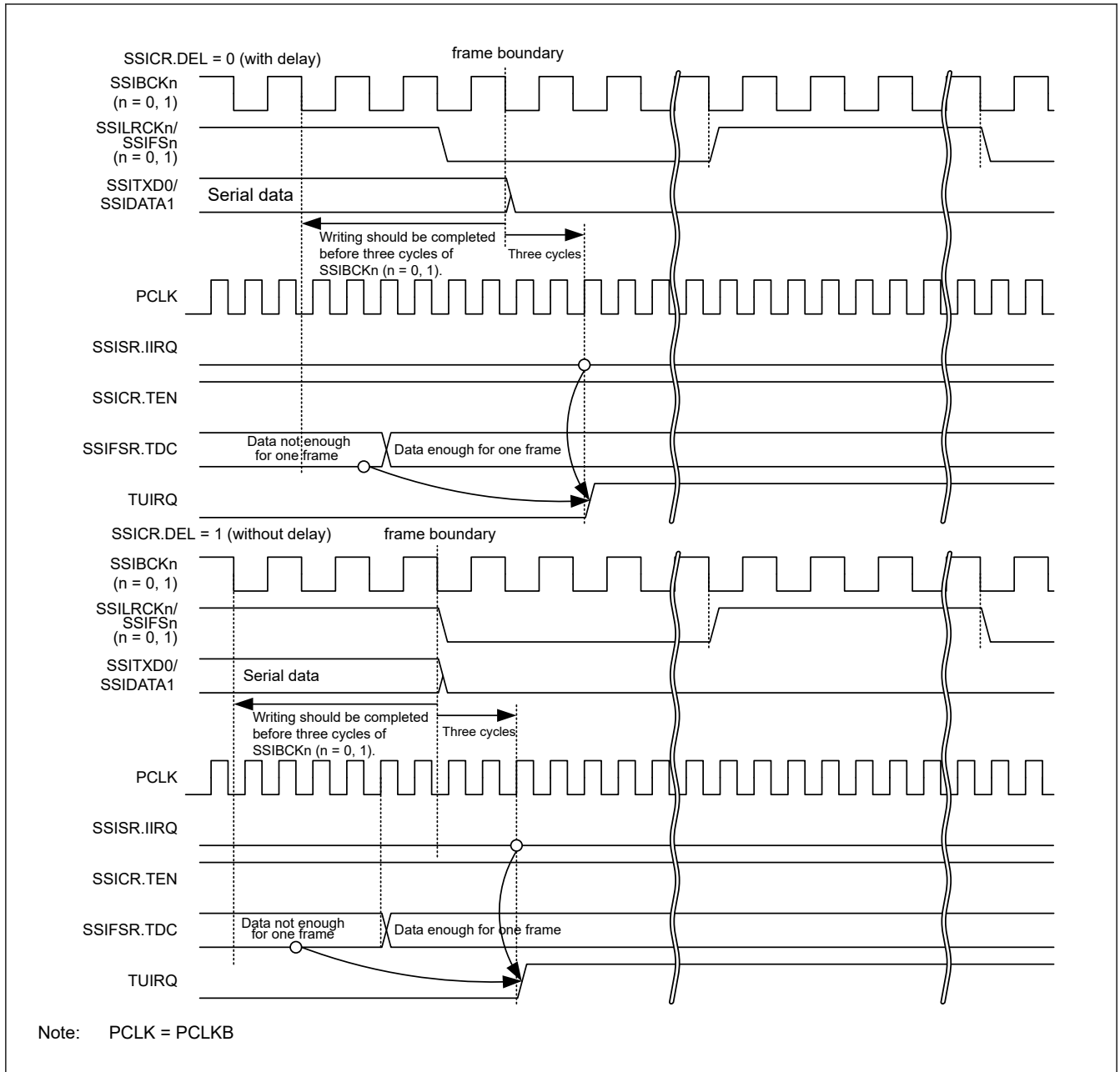


Figure 37.20 TUIRQ setting timing (when communication continues)

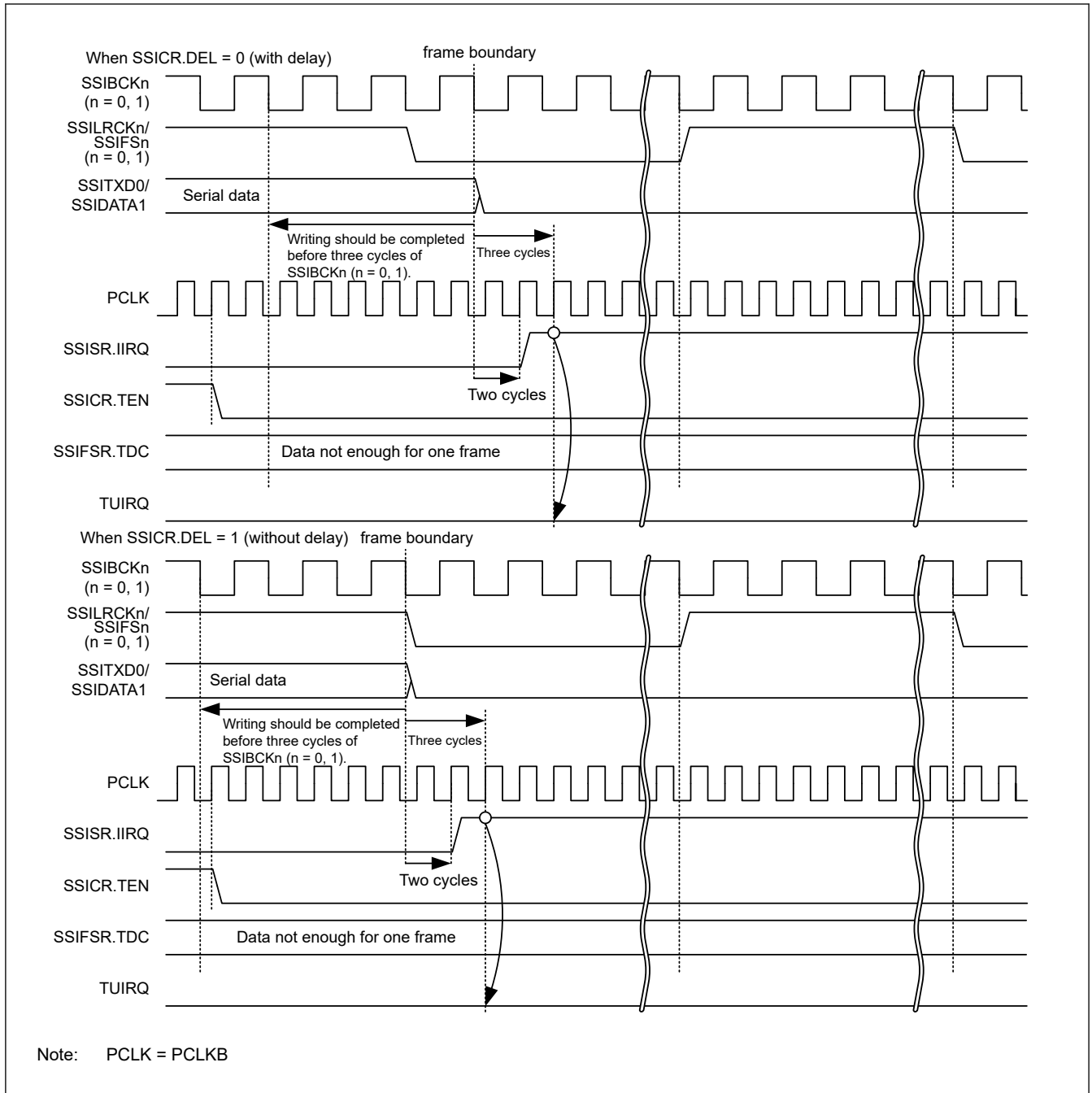


Figure 37.21 TUIRQ setting timing (when communication stops)

### 37.2.3 SSIFCR : FIFO Control Register

Base address: SSIE<sub>n</sub> = 0x4025\_D000 + 0x0100 × n (n = 0, 1)  
 SSIE<sub>n</sub>\_NS = 0x5025\_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AUCK E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSIRS T
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	BSW	—	—	—	—	—	—	—	TIE	RIE	TFRS T	RFRS T
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFRST	Receive FIFO Data Register Reset* <sup>1</sup> 0: Clears a receive data FIFO reset condition 1: Sets a receive data FIFO reset condition	R/W
1	TFRST	Transmit FIFO Data Register Reset* <sup>1</sup> 0: Clears a transmit data FIFO reset condition 1: Sets a transmit data FIFO reset condition	R/W
2	RIE	Receive Data Full Interrupt Output Enable 0: Disables receive data full interrupts 1: Enables receive data full interrupts	R/W
3	TIE	Transmit Data Empty Interrupt Output Enable 0: Disables transmit data empty interrupts 1: Enables transmit data empty interrupts	R/W
10:4	—	These bits are read as 0. The write value should be 0.	R/W
11	BSW	Byte Swap Enable* <sup>1</sup> 0: Disables byte swap 1: Enables byte swap	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
16	SSIRST	Software Reset 0: Clears a software reset condition 1: Sets a software reset condition	R/W
30:17	—	These bits are read as 0. The write value should be 0.	R/W
31	AUCKE	AUDIO_MCK Enable in Mastermode Communication* <sup>1</sup> 0: Disables supply of AUDIO_MCK 1: Enables supply of AUDIO_MCK	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Writing to these bits while SSIE is in a communication state (SSISR.IIRQ = 0) is prohibited. If the value of these bits is changed by rewriting, subsequent operation is unpredictable.

This register sets a software reset, byte swap, and enable/disable of interrupt requests.

#### RFRST bit (Receive FIFO Data Register Reset)

The RFRST bit sets a software reset of the receive FIFO data register (SSIFRDR). Writing 1 to this bit initializes the internal state related to SSIFRDR. The register bits subject to the software reset triggered by this bit are indicated by shading in [Table 37.7](#). Because this bit is not automatically cleared after it has been set, write 0 to this bit to release the register bits from the software reset. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedural step.

This bit is subject to the software reset by the SSIRST bit. Because the software reset by the SSIRST bit has priority over the reset by this bit, setting this bit is ignored when the SSIRST bit is set.

**Table 37.7 Bits subject to software reset by the RFRST bit**

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	0x00	+0	—	CKS	TUI EN	TOI EN	RUI EN	ROI EN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCK P	LRC KP	SPD P	SDT A	PDT A	DEL	CKDV[3:0]			MU EN	—	TEN	REN	
SSISR	0x04	+0	—	—	TUI RQ	TOI RQ	RUI RQ	ROI RQ	IIRQ	—	—	—	—	—	—	—	—	
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
SSIFCR	0x10	+0	AUC KE	—	—	—	—	—	—	—	—	—	—	—	—	—	SSI RST	
		+2	—	—	—	—	BS W	—	—	—	—	—	—	—	TIE	RIE	TFR ST	RFR ST
SSIFSR	0x14	+0	—	—	TDC[5:0]					—	—	—	—	—	—	—	TDE	
		+2	—	—	RDC[5:0]					—	—	—	—	—	—	—	RDF	
SSIFTDR	0x18	+0	SSIFTDR[31:16]															
		+2	SSIFTDR[15:0]															
SSIFRDR	0x1C	+0	SSIFRDR[31:16]															
		+2	SSIFRDR[15:0]															
SSIOFR	0x20	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
		+2	—	—	—	—	—	—	BCK AST P	LRC ONT	—	—	—	—	—	—	—	OMOD[1:0]
SSISCR	0x24	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

**TFRST bit (Transmit FIFO Data Register Reset)**

The TFRST bit sets a software reset of the transmit FIFO data register (SSIFTDR). Writing 1 to this bit initializes the internal state related to SSIFTDR. The register bits subject to the software reset triggered by this bit are indicated by shading in Table 37.8. Because this bit is not automatically cleared after it has been set, write 0 to this bit to release the register bits from the software reset. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedural step.

This bit is subject to the software reset by the SSIRST bit. Because the software reset by the SSIRST bit has priority over the reset by this bit, setting this bit is ignored when the SSIRST bit is set.

**Table 37.8 Bits subject to software reset by the TFRST bit (1 of 2)**

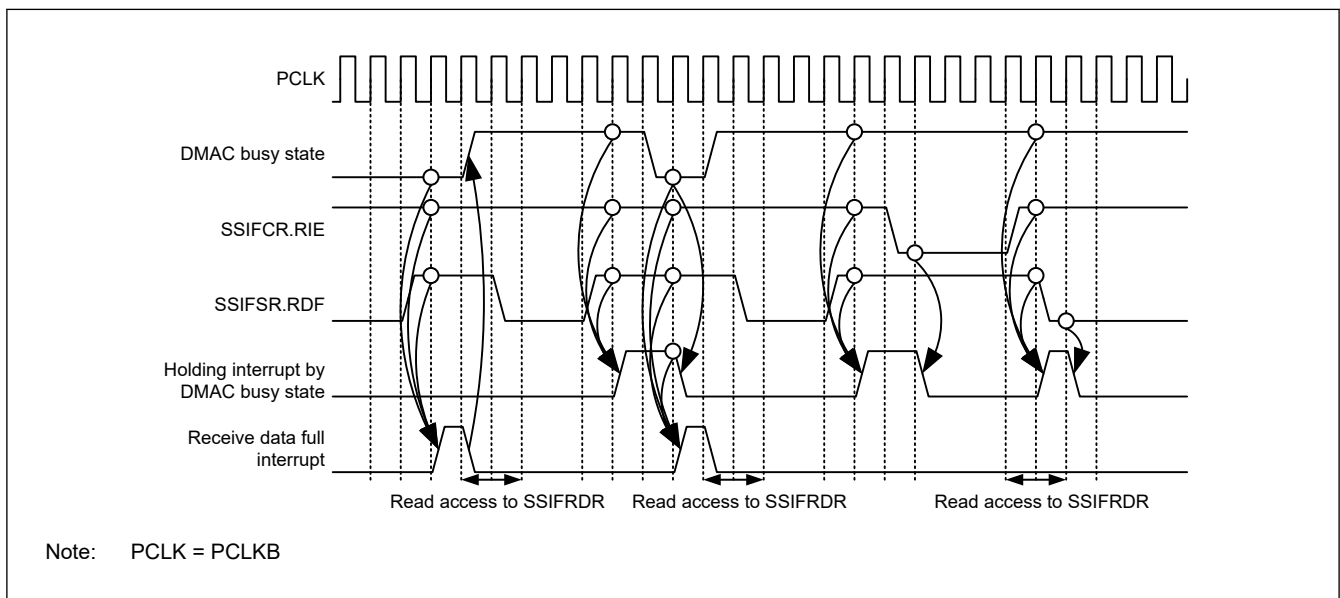
Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	0x00	+0	—	CKS	TUI EN	TOI EN	RUI EN	ROI EN	IIEN	—	FRM[1:0]		DWL[2:0]			SWL[2:0]		
		+2	—	MST	BCK P	LRC KP	SPD P	SDT A	PDT A	DEL	CKDV[3:0]			MU EN	—	TEN	REN	
SSISR	0x04	+0	—	—	TUI RQ	TOI RQ	RUI RQ	ROI RQ	IIRQ	—	—	—	—	—	—	—	—	
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	

**Table 37.8 Bits subject to software reset by the TFRST bit (2 of 2)**

Symbol	Address (BASE+)	+0								+1							
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSIFCR	0x10	+0	AUC KE	—	—	—	—	—	—	—	—	—	—	—	—	—	SSI RST
		+2	—	—	—	—	BS W	—	—	—	—	—	—	TIE	RIE	TFR ST	RFR ST
SSIFSR	0x14	+0	—	—	TDC[5:0]					—	—	—	—	—	—	—	TDE
		+2	—	—	RDC[5:0]					—	—	—	—	—	—	—	RDF
SSIFTDR	0x18	+0	SSIFTDR[31:16]														
		+2	SSIFTDR[15:0]														
SSIFRDR	0x1C	+0	SSIFRDR[31:16]														
		+2	SSIFRDR[15:0]														
SSIOFR	0x20	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	—	—	—	BCK AST P	LRC ONT	—	—	—	—	—	—	OMOD[1:0]
SSISCR	0x24	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]			

**RIE bit (Receive Data Full Interrupt Output Enable)**

The RIE bit enables/disables output of receive data full interrupts. Use a receive data full interrupt as an interrupt to trigger data reading from the receive FIFO data register. Write 1 to this bit after specifying the setting condition for receive data full interrupt (by using the SSISCR.RDFS bit). [Figure 37.22](#) shows the timing of generating the receive data full interrupt.



**Figure 37.22 Timing of receive data full interrupt**

**TIE bit (Transmit Data Empty Interrupt Output Enable)**

The TIE bit enables/disables output of transmit data empty interrupts. Use a transmit data empty interrupt as an interrupt to trigger data writing to the transmit FIFO data register. Write 1 to this bit after specifying the setting condition for transmit data empty interrupt (by using the SSISCR.TDES bit). [Figure 37.23](#) shows the timing of generating the transmit data empty interrupt.

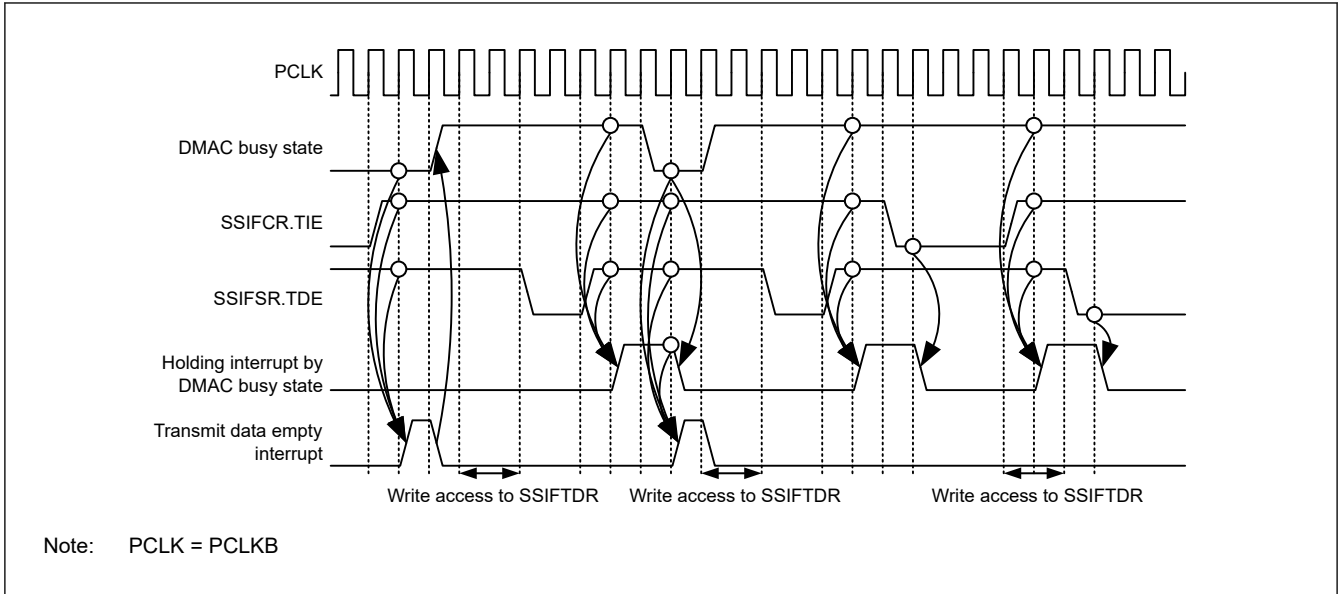


Figure 37.23 Timing of transmit data empty interrupt

**BSW bit (Byte Swap Enable)**

The BSW bit enables/disables byte swap of register access for the transmit FIFO data register (SSIFTDR) and the receive FIFO data register (SSIFRDR). This bit is valid only with 16-bit access or 32-bit access to SSIFTDR and SSIFRDR. For details, see [Figure 37.24](#).

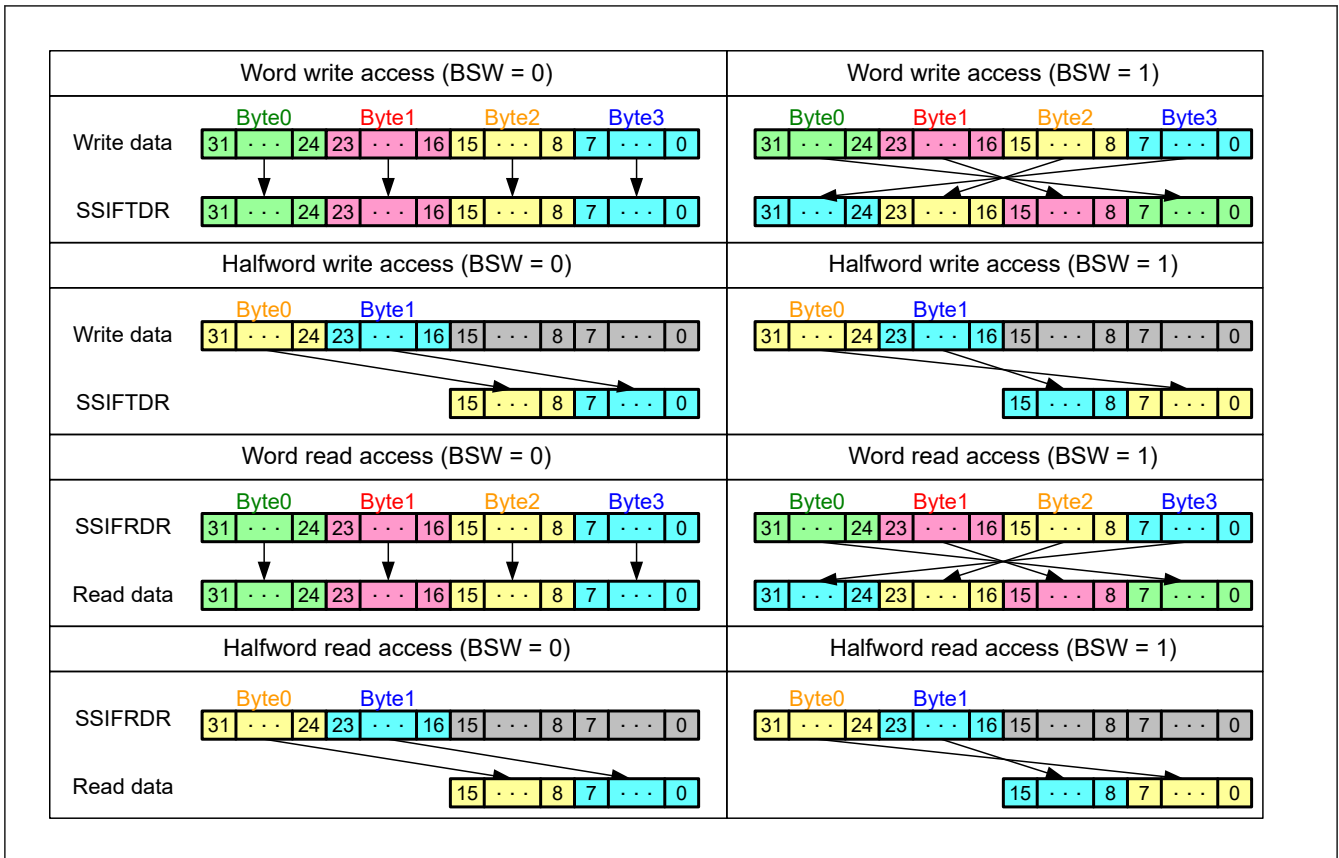


Figure 37.24 Operation example of byte swap

**SSIRST bit (Software Reset)**

The SSIRST bit sets a software reset of SSIE. Writing 1 to this bit initializes the internal state of SSIE. The register bits subject to the software reset triggered by this bit are indicated by shading in [Table 37.9](#). Because this bit is not automatically

cleared after it has been set, write 0 to this bit to release the register bits from the software reset. After writing 0 to this bit, be sure to check that this bit is 0 before starting the next procedural step.

To stop communication of SSIE immediately, after turning off the peripheral functions, write 1 to this bit. Initialization by a software reset is performed without any relation with the bit clock.

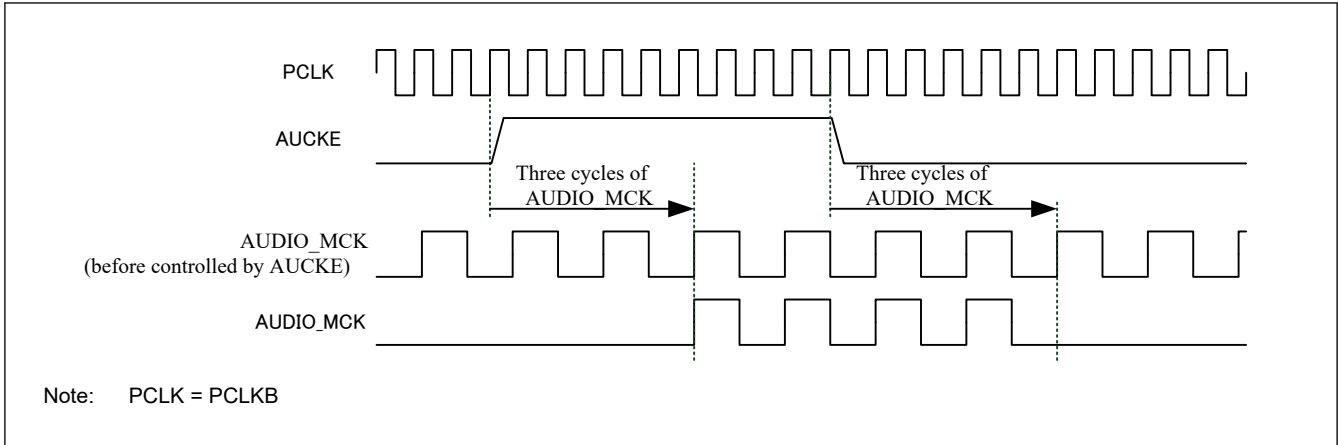
**Table 37.9 Bits subject to software reset by the SSIRST bit**

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	0x00	+0	—	CKS	TUI EN	TOI EN	RUI EN	ROI EN	IEN	—	FRM[1:0]	DWL[2:0]			SWL[2:0]			
		+2	—	MST	BCK P	LRC KP	SPD P	SDT A	PDT A	DEL	CKDV[3:0]			MU EN	—	TEN	REN	
SSISR	0x04	+0	—	—	TUI RQ	TOI RQ	RUI RQ	ROI RQ	IIRQ	—	—	—	—	—	—	—	—	
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
SSIFCR	0x10	+0	AUC KE	—	—	—	—	—	—	—	—	—	—	—	—	—	SSI RST	
		+2	—	—	—	—	BS W	—	—	—	—	—	—	—	TIE	RIE	TFR ST	RFR ST
SSIFSR	0x14	+0	—	—	TDC[5:0]					—	—	—	—	—	—	—	TDE	
		+2	—	—	RDC[5:0]					—	—	—	—	—	—	—	RDF	
SSIFTDR	0x18	+0	SSIFTDR[31:16]															
		+2	SSIFTDR[15:0]															
SSIFRDR	0x1C	+0	SSIFRDR[31:16]															
		+2	SSIFRDR[15:0]															
SSIOFR	0x20	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
		+2	—	—	—	—	—	—	BCK AST P	LRC ONT	—	—	—	—	—	—	OMOD[1:0]	
SSISCR	0x24	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

**AUCKE bit (AUDIO\_MCK Enable in Mastermode Communication)**

The AUCKE bit enables/disables supply to AUDIO\_MCK while in master-mode communication (MST = 1).

Changing the value of this bit must be performed only after specifying the settings related to AUDIO\_MCK (by using the CKS, MST, BCKP, and CKDV bits in the SSICR register).

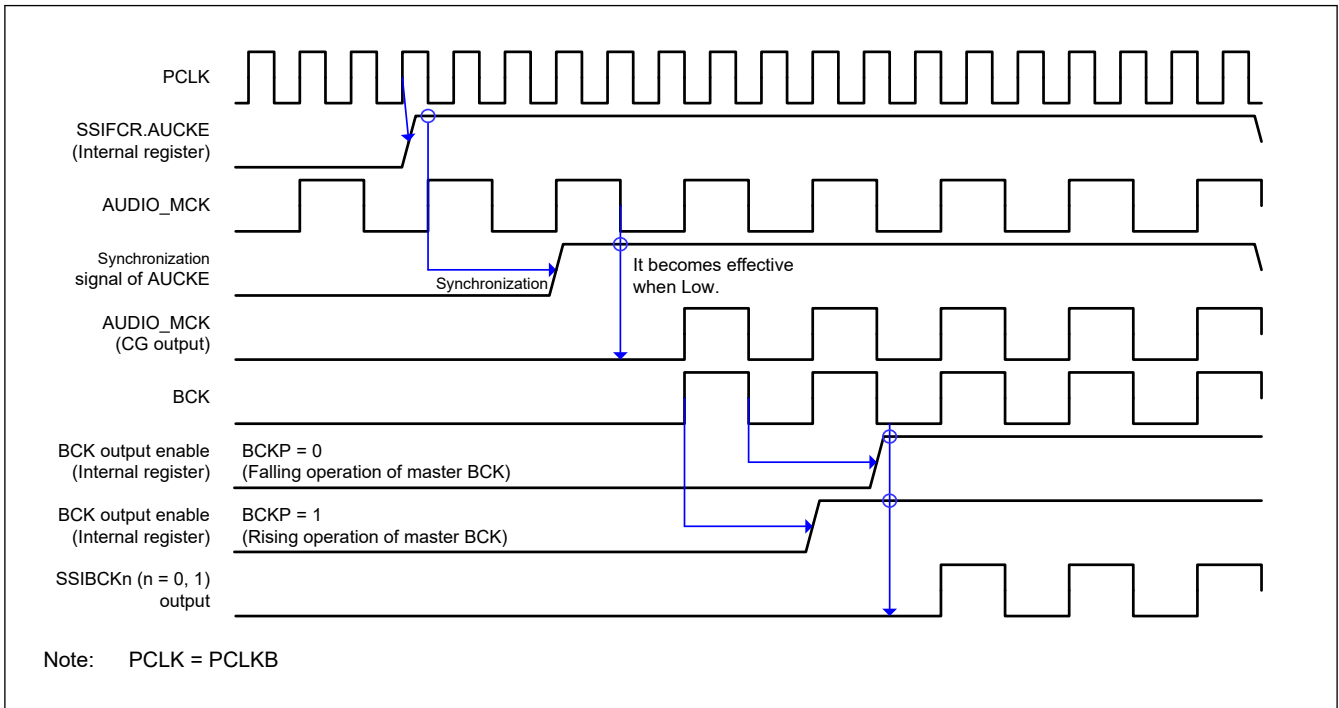


**Figure 37.25 Stop/resume of AUDIO\_MCK**

Note: In slave-mode communication (SSICR.MST = 0), SSIE needs supply of SSIBCKn (n = 0, 1). To stop BCK on the master side, make sure that SSIE is in the idle state (SSISR.IIRQ = 1). If BCK is stopped before SSIE becomes idle, take the procedure to start communication in [Figure 37.52](#) or wait for an idle state by taking the procedure to resume communication in [Figure 37.58](#).

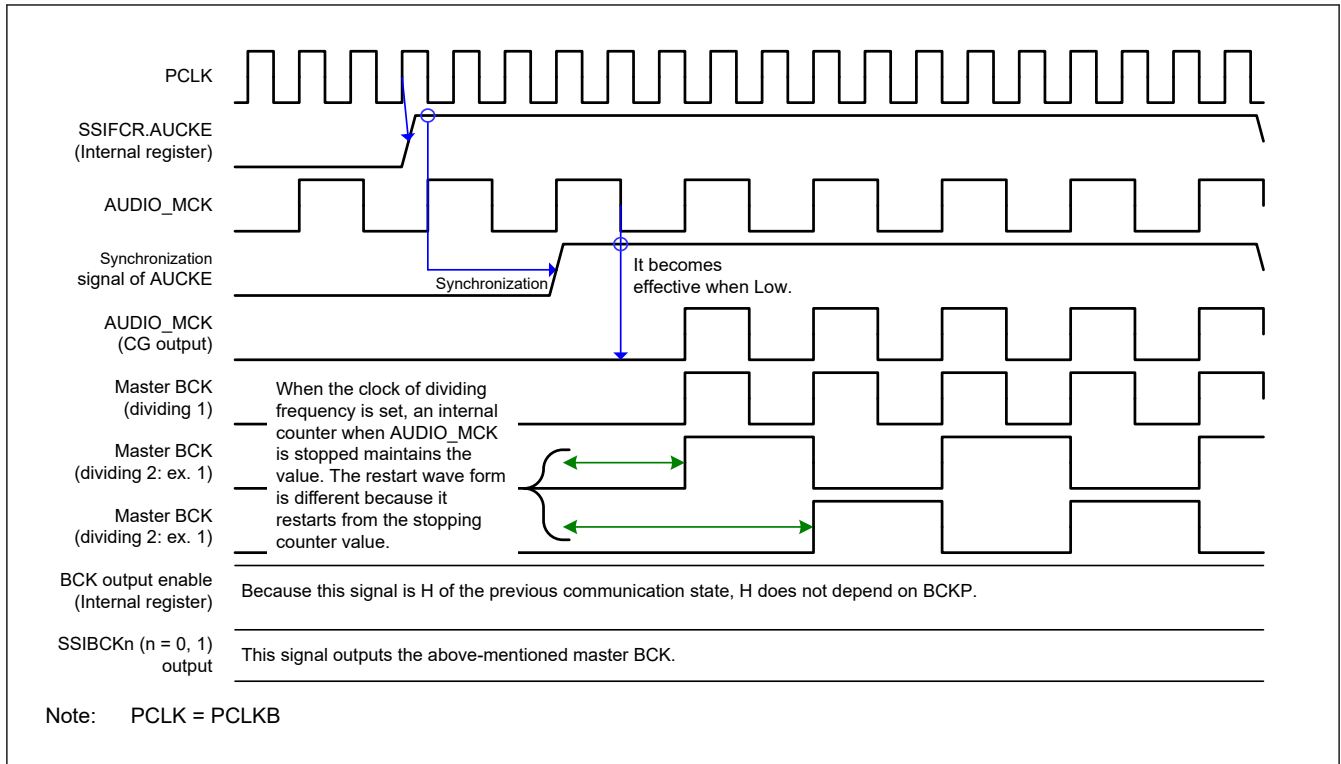
In master-mode communication (SSICR.MST = 1), SSIE operates with the audio clock (AUDIO\_MCK). To stop SSIE completely, make sure that SSIE is in the idle state (SSISR.IIRQ = 1) and then write 0 to SSIFCR.AUCKE. If 0 is written to SSIFCR.AUCKE before SSIE becomes idle, take the procedure to start communication in [Figure 37.52](#).

[Figure 37.26](#) and [Figure 37.27](#) show the timings of signal operation in the period from setting this bit to 1 to the output to the SSIBCKn (n = 0, 1) pin.



**Figure 37.26 Timing diagram for the operation from system reset to start of master-mode communication**





**Figure 37.27** Timing diagram for the operation from stop of communication to start of master-mode communication

**Note:** If the supply of AUDIO\_MCK stops, the value of the SSIBCKn (n = 0, 1) pin is held. Therefore, the SSIBCKn (n = 0, 1) signal might stop in the H (high level) state.

### 37.2.4 SSIFSR : FIFO Status Register

Base address: SSIE<sub>n</sub> = 0x4025\_D000 + 0x0100 × n (n = 0, 1)  
 SSIE<sub>n</sub>\_NS = 0x5025\_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Bit field:	—	—	TDC[5:0]					—	—	—	—	—	—	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit field:	—	—	RDC[5:0]					—	—	—	—	—	—	—	—	—	RDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Symbol	Function	R/W
0	RDF	Receive Data Full Flag 0: The size of received data in SSIFRDR is not more than the value of SSISCR.RDFS. 1: The size of received data in SSIFRDR is not less than the value of SSISCR.RDFS plus one.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
13:8	RDC[5:0]	Receive Data Count Number of valid data stored in the receive FIFO data register	R
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	TDE	Transmit Data Empty Flag 0: The free space of SSIFTDR is not more than the value of SSISCR.TDES. 1: The free space of SSIFTDR is not less than the value of SSISCR.TDES plus one.	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
29:24	TDC[5:0]	Transmit Data Count Number of valid data stored in the transmit FIFO data register	R
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register is configured with status flags that indicate the status of the transmit FIFO data register and the receive FIFO data register.

### RDF flag (Receive Data Full Flag)

The RDF flag indicates that the receive FIFO data register (SSIFRDR) has unread received data not less than the amount set with the SSISCR.RDFS bits plus one. This flag is set by automatic determination but it must be cleared by register access.

[Priority order for setting and clearing]

Clearing is prioritized.

[Clearing condition]

Either of the following two:<sup>\*1</sup>

1. Writing 0 to this bit after reading 1 from this bit (CPU operation)<sup>\*2</sup>
2. Last access (DTC/DMAC operation) to read data from SSIFRDR by an interrupt routine using the DTC and DMAC.

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in [Figure 37.19](#))
2. After the PCLKB cycle in which the last access instruction is issued to read data from SSIFRDR by an interrupt routine using the DTC and DMAC.

[Setting condition]

SSIFRDR has free space not less than the amount set with the SSISCR.RDFS bits plus one.

[Setting timing]

At completion of transfer from the shift register that results in SSIFRDR having data not less than the amount set with the SSISCR.RDFS bits plus one.

Note 1. These bits are cleared by a software reset (SSIFCR.SSIRST = 1) and receive FIFO data register reset (SSIFCR.RFRST = 1). Reset conditions available for these bits are the software reset and receive FIFO data register reset as well as the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following four conditions is met:

- A software reset is done (SSIFCR.SSIRST = 1).
- A receive FIFO data register reset is done (SSIFCR.RFRST = 1).
- After 1 has been read, writing of 0 is complete.
- Last access is performed to read data from SSIFRDR by an interrupt routine using the DTC and DMAC.

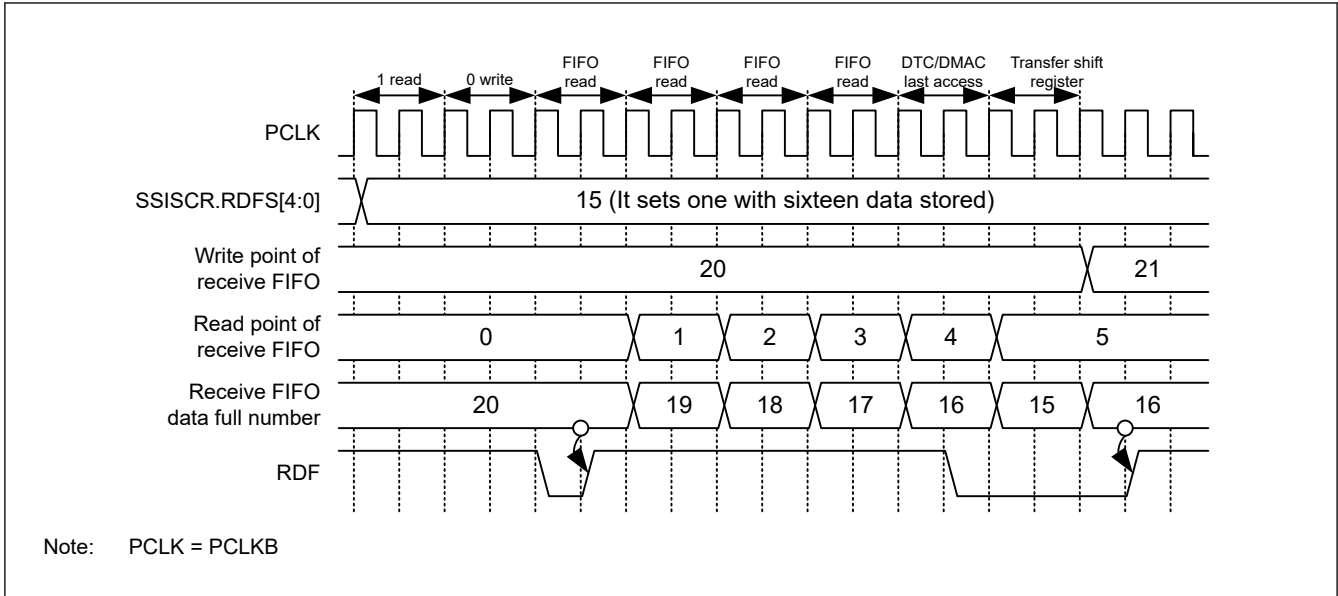


Figure 37.28 Timing diagram for setting and clearing RDF

**RDC[5:0] bits (Receive Data Count)**

The RDC[5:0] bits indicate the number of valid data that are stored in the receive FIFO data register (SSIFRDR). With these bits as 0x00, there is no received data. With 0x20, the register is filled with received data and there is no free space.

**TDE flag (Transmit Data Empty Flag)**

The TDE flag indicates that the transmit FIFO data register (SSIFTDR) has free space not less than the amount set with the SSISCR.TDES bits plus one. This flag is set by automatic determination but it must be cleared by register access.

[Priority order for setting and clearing]

Clearing is prioritized.\*1

[Clearing condition]

Either of the following two:

1. Writing 0 to this bit after reading 1 from this bit (CPU operation)\*2
2. Last access (DTC/DMAC operation) to write data to SSIFTDR by an interrupt routine using the DTC and DMAC.

[Clearing timing]

Clearing timing corresponding to the above clearing condition

1. When 0 is written to this bit after reading 1 from this bit (same as the timing in Figure 37.19)
2. Last access (DTC/DMAC operation) to write data to SSIFTDR by an interrupt routine using the DTC and DMAC.

[Setting condition]

SSIFTDR has free space not less than the amount set with the SSISCR.TDES bits plus one.

[Setting timing]

While operating on PCLKB, SSIFTDR is found to have free space not less than “size set in the SSISCR.TDES bits + 1.”

Note 1. This bit is cleared by a software reset (SSIFCR.SSIRST = 1) and transmit FIFO data register reset (SSIFCR.TFRST = 1). The software reset and transmit FIFO data register reset have priority over all the clearing conditions described above.

Note 2. After reading 1 from this bit, this bit is cleared when one of the following four conditions is met:

- A software reset is done (SSIFCR.SSIRST = 1).
- A transmit FIFO data register reset is done (SSIFCR.TFRST = 1).
- After 1 has been read, writing of 0 is complete.
- Last access is performed to write data to SSIFTDR by an interrupt routine using the DTC and DMAC.

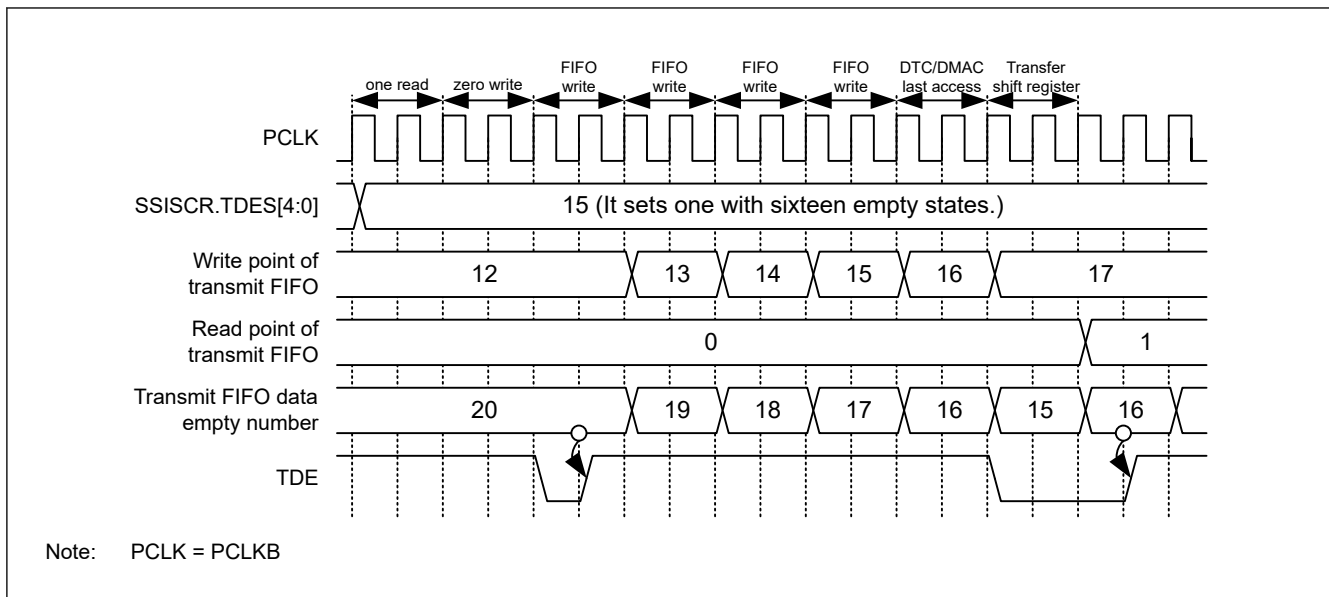


Figure 37.29 Timing diagram for setting and clearing TDE

TDC[5:0] bits (Transmit Data Count)

The TDC[5:0] bits indicate the number of valid data that are stored in the transmit FIFO data register (SSIFTDR). With these bits as 0x00, there is no data to be transmitted. With 0x20, there is no space to write data.

37.2.5 SSIFTDR : Transmit FIFO Data Register

Base address: SSIE<sub>n</sub> = 0x4025\_D000 + 0x0100 × n (n = 0, 1)  
 SSIE<sub>n</sub>\_NS = 0x5025\_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x18

Bit position: 31

0

Bit field: SSIFTDR[31:0]

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	SSIFTDR[31:0]	Transmit FIFO Data	W

Note: S-TYPE-3, P-TYPE-3

This register stores data to be serially transmitted. 0 is returned when this register is read.

When you use this register for transmission, specify data writing to this register as the DTC/DMAC operation that is triggered by a transmit data empty interrupt. Determine the access size to this register according to the data word length to be communicated in Table 37.10.

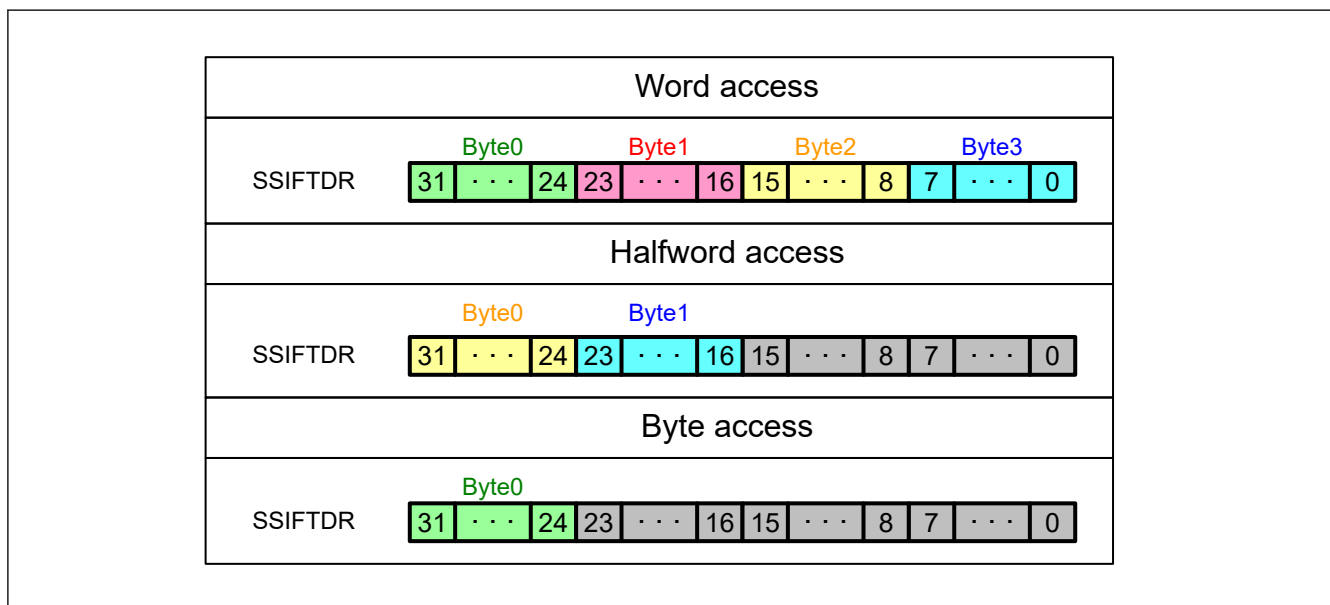
Table 37.10 Register access restriction to FIFOs (1 of 2)

Access Size		Byte	Halfword	Word
SSICR.DWL[2:0]	Data Word Length			
000b	8	✓	—	—
001b	16	—	✓	—
010b	18	—	—	✓
011b	20	—	—	✓
100b	22	—	—	✓
101b	24	—	—	✓

**Table 37.10 Register access restriction to FIFOs (2 of 2)**

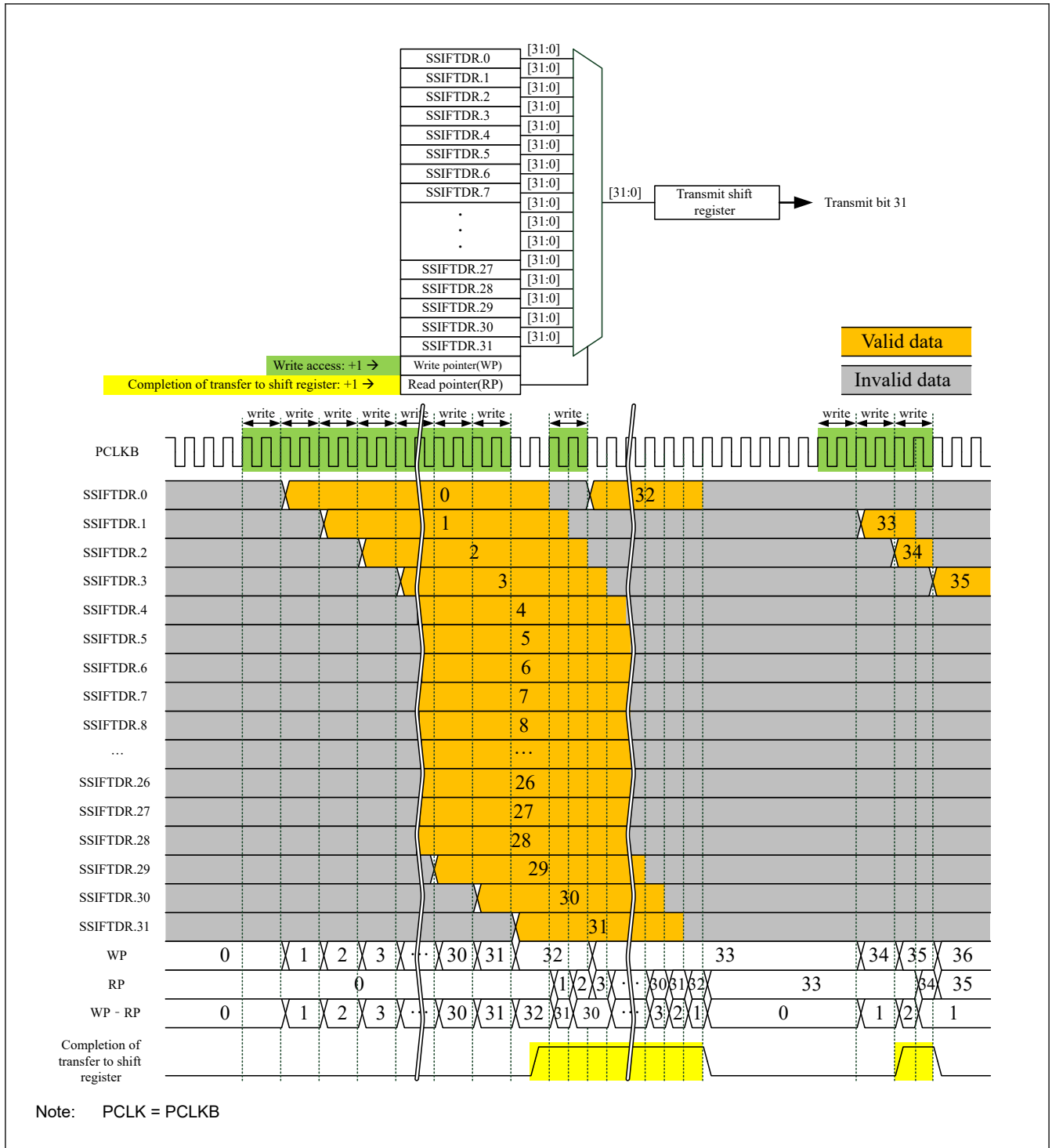
Access Size		Byte	Halfword	Word
SSICR.DWL[2:0]	Data Word Length			
110b	32	—	—	✓
111b	Setting prohibited	—	—	—

Figure 37.30 shows register access to the transmit FIFO data register.



**Figure 37.30 Example of register access to the transmit FIFO data register**

Figure 37.31 shows the configurations and operation examples of the transmit FIFO data register and transmit shift register. The configurations are for storing data to FIFO and not related with communication.



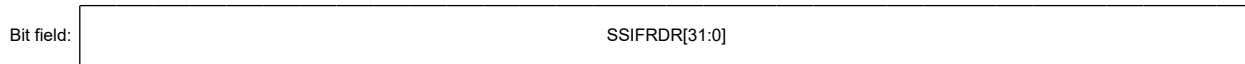
**Figure 37.31 Configuration of the transmit FIFO data register and transmit shift register, and FIFO operation example**

### 37.2.6 SSIFRDR : Receive FIFO Data Register

Base address:  $SSIE_n = 0x4025\_D000 + 0x0100 \times n \ (n = 0, 1)$   
 $SSIE\_NS = 0x5025\_D000 + 0x0100 \times n \ (n = 0, 1)$

Offset address: 0x1C

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	SSIFRDR[31:0]	Receive FIFO Data	R

Note: S-TYPE-3, P-TYPE-3

When you use this register for reception, specify data reading from this register as the DTC/DMAC operation that is triggered by a transmit data empty interrupt. Determine the access size to this register according to the data word length to be communicated in [Table 37.10](#).

Register access to the receive FIFO data register is same as for the transmit FIFO data register.

[Figure 37.32](#) shows the configurations and operation examples of the receive FIFO data register and receive shift register.

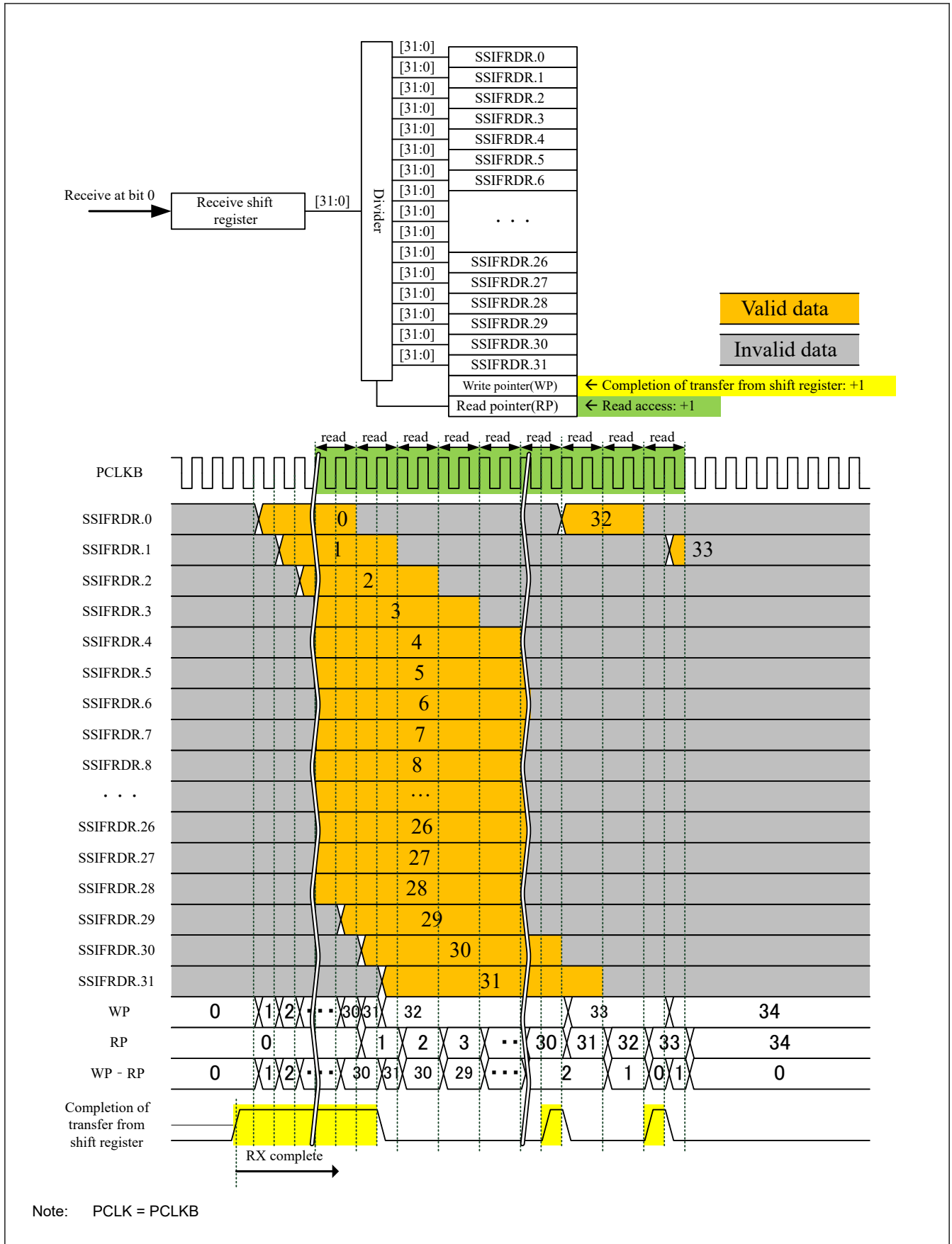


Figure 37.32 Configuration of the receive FIFO data register and receive shift register, and FIFO operation example



### 37.2.7 SSIOFR : Audio Format Register

Base address: SSIE<sub>n</sub> = 0x4025\_D000 + 0x0100 × n (n = 0, 1)  
 SSIE<sub>n</sub>\_NS = 0x5025\_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	BCKA STP	LRCO NT	—	—	—	—	—	—	OMOD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OMOD[1:0]	Audio Format Select* <sup>3</sup> * <sup>4</sup> 0 0: I <sup>2</sup> S format 0 1: TDM format 1 0: Monaural format 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	LRCONT	Whether to Enable LRCK/FS Continuation* <sup>1</sup> * <sup>2</sup> 0: Disables LRCK/FS continuation 1: Enables LRCK/FS continuation	R/W
9	BCKASTP	Whether to Enable Stopping BCK Output When SSIE is in Idle Status* <sup>1</sup> * <sup>2</sup> 0: Always outputs BCK to the SSIBCK <sub>n</sub> (n = 0, 1) pin 1: Automatically controls output of BCK to the SSIBCK <sub>n</sub> (n = 0, 1) pin	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. This bit is valid only in master-mode communication (SSICR.MST = 1). The setting is invalid in slave-mode communication (SSICR.MST = 0).

Note 2. The BCKASTP and LRCONT bits must not be set to 1 together.

Note 3. While SSIE is communicating (SSISR.IIRQ = 0), writing to these bits is prohibited. If the value of these bits is changed by writing, subsequent operation is unpredictable.

Note 4. If the communication format of other-party device is compatible with a communication format of SSIE, specify and use the communication format that enables communication with the other-party device.

This register is used to set an audio format (which involves the settings of communication format, LR clock/frame synchronization continuation mode, and BCK output stop).

#### OMOD[1:0] bits (Audio Format Select)

The OMOD[1:0] bits set an audio format. Writing to these bits must be performed when the LR clock supply to the SSILRCK<sub>n</sub>/SSIFS<sub>n</sub> (n = 0, 1) pin is stopped. For details about the output of LR clock, see the detailed description of the LRCONT bit in [section 37.2.7. SSIOFR : Audio Format Register](#).

#### LRCONT bit (Whether to Enable LRCK/FS Continuation)

The LRCONT bit enables or disables the output from SSILRCK<sub>n</sub>/SSIFS<sub>n</sub> (n = 0, 1) pin when the communication mode is master-mode communication (SSICR.MST = 1) and SSIE is in the idle state (SSISR.IIRQ = 1).

Even in the idle state, a signal can output from the SSILRCK<sub>n</sub>/SSIFS<sub>n</sub> (n = 0, 1) pin when this bit is set to 1 (to enable LR clock/frame synchronization continuation) in master mode (SSICR.MST = 1).

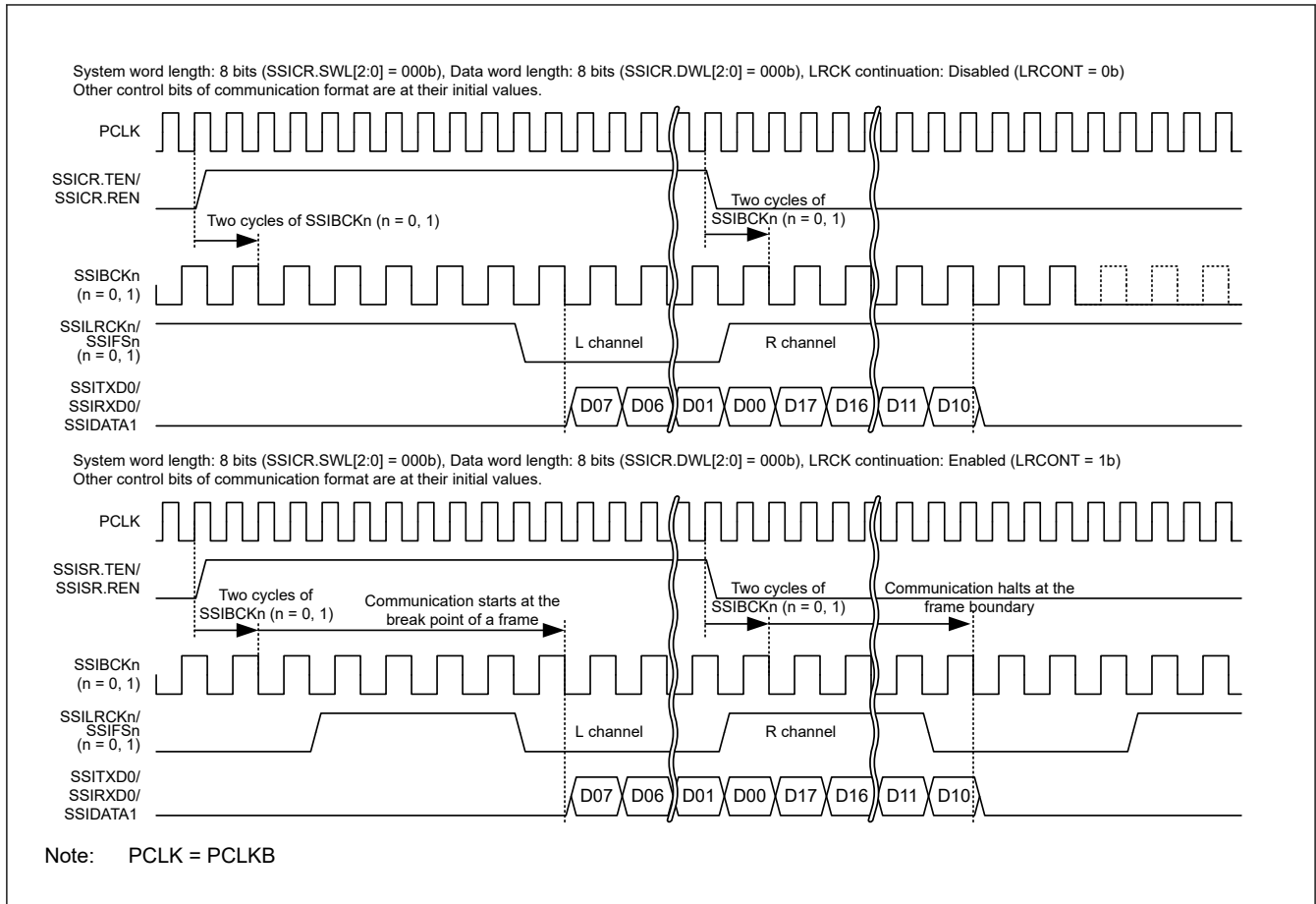


Figure 37.33 Example of LR clock/frame synchronization continuation operation

**BCKASTP bit (Whether to Enable Stopping BCK Output When SSIE is in Idle Status)**

The BCKASTP bit turns on or off the function to output BCK to the SSIBCKn (n = 0, 1) pin according to the communication shown in Figure 37.34 and Figure 37.35 in master-mode communication (SSICR.MST = 1).

Changing the value of this bit must be performed only after setting the communication format to be used.

This bit must be used in the following way:

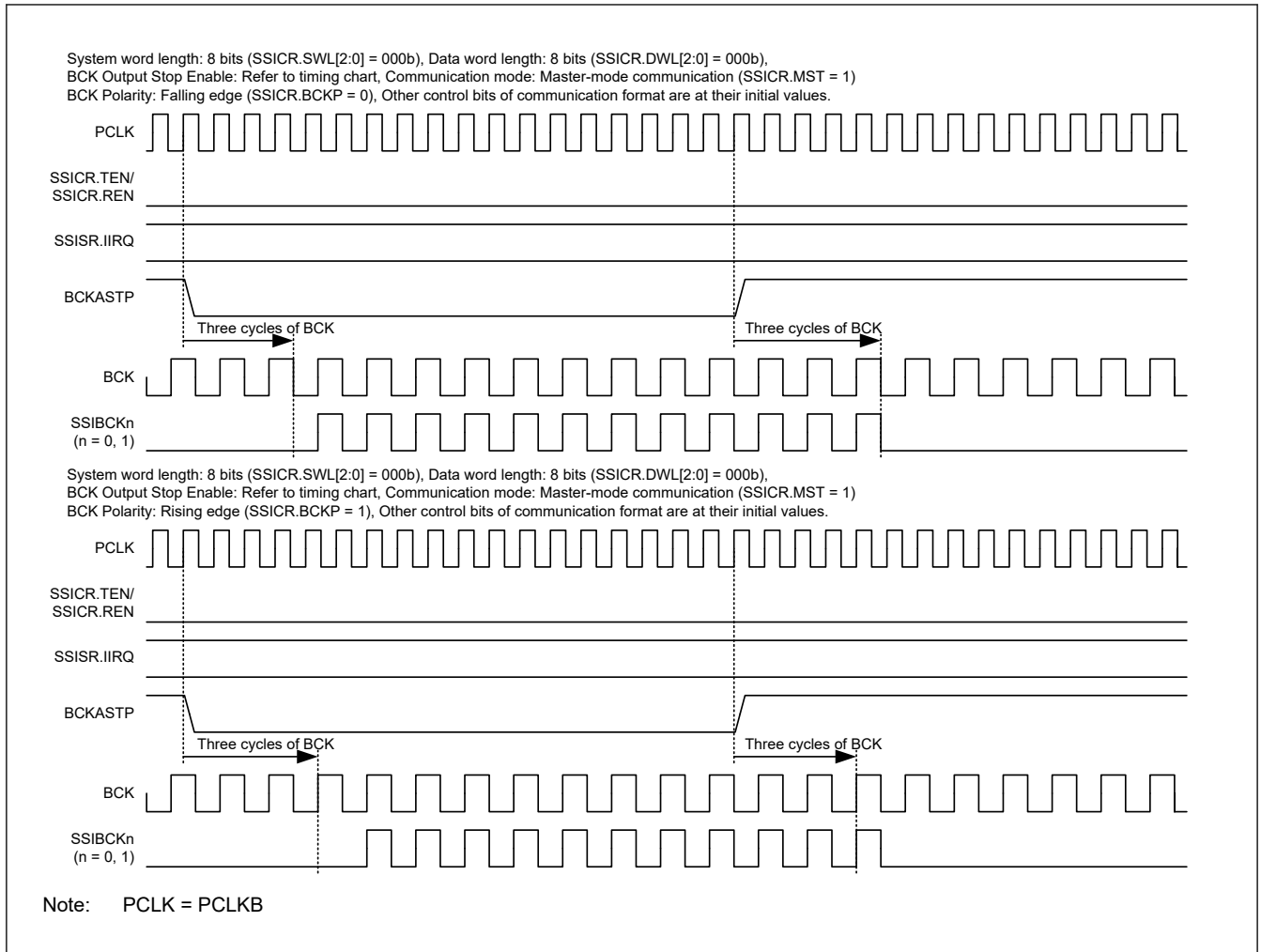
Write 0 to the BCKASTP bit, and then start communication. During the communication, write 1 to the BCKASTP bit. By this operation, the bit clock output to the SSIBCKn (n = 0, 1) pin stops automatically when the communication stops. To resume the communication, set SSIE to the idle state (SSICR.IIRQ = 1), enable the supply of AUDIO\_MCK (SSIFCR.AUCKE = 1), and then write 0 to the BCKASTP bit.

When the communication mode is master-mode communication (SSICR.MST = 1) and SSIE is in the idle state (SSICR.IIRQ = 1):

Table 37.11 BCKASTP bit status and SSIBCKn (n = 0, 1) pin output

BCKASTP Bit	SSIBCKn (n = 0, 1) Pin Output Status
0	Output
1	Stopped

Note: The BCKASTP bit cannot be used when the other-party device (which is a slave) requires the clock output from the SSIBCKn (n = 0, 1) pin before and during communication. In such a case, use the BCKASTP bit to stop the clock only after communication. For the timing of enabling the clock stop function, see Figure 37.34.



**Figure 37.34 Example operation of the BCKASTP bit (idle state)**

When the communication mode is master-mode communication (SSICR.MST = 1) and the BCK output stop function is enabled (BCKASTP = 1):

Details of the BCK output to the SSIBCKn (n = 0, 1) pin are as follows:

Output start timing: BCK is output in appropriate timing so that a valid edge is generated when the LR clock/frame synchronization signal shifts to a valid value.

Output stop timing: 1 to 1.5 clock cycles after a frame boundary.

For details about the timings, see the timing diagram in [Figure 37.35](#).

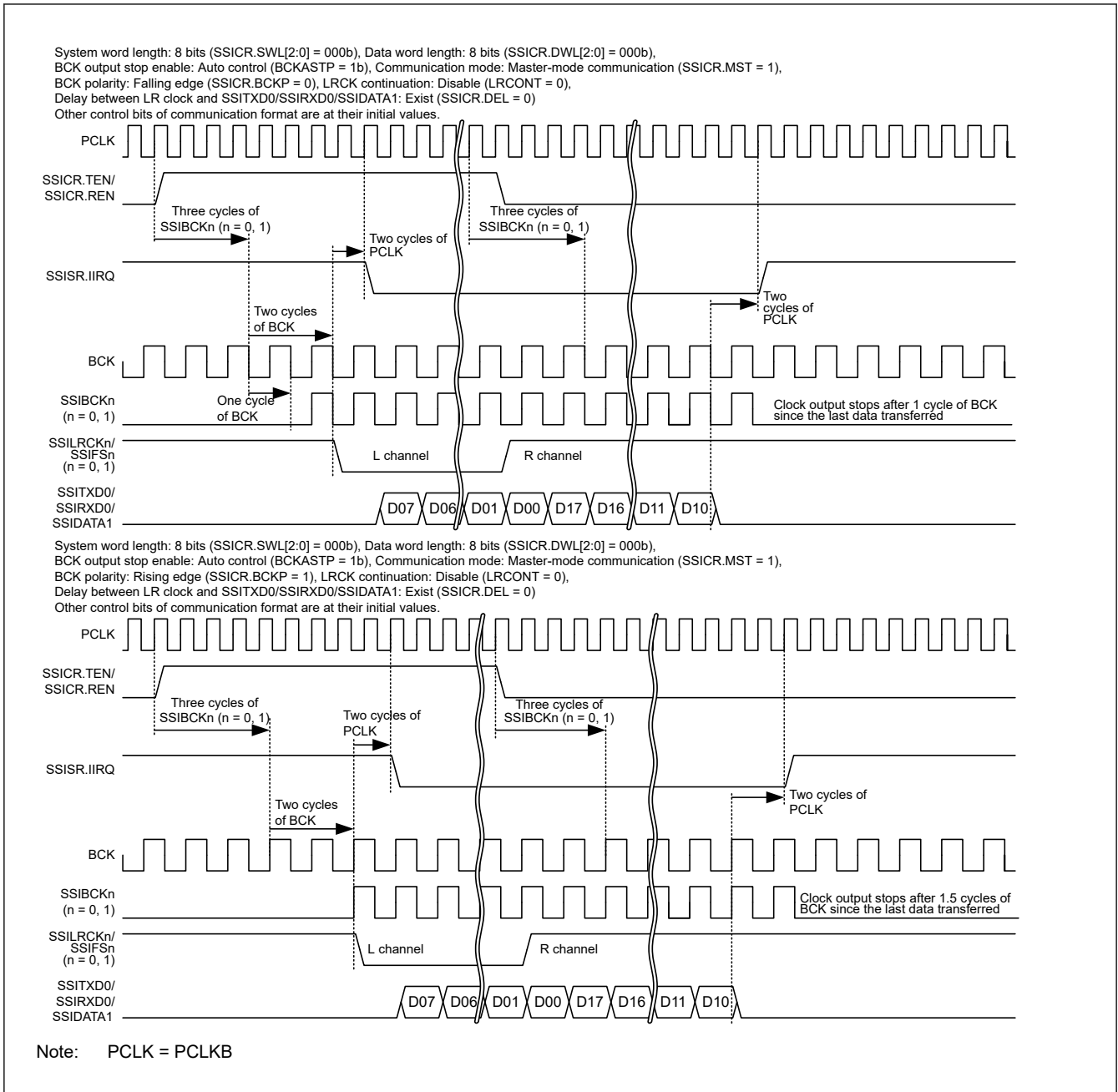


Figure 37.35 Example operation of the BCKASTP bit (communication operation with BCKASTP = 1)

### 37.2.8 SSISCR : Status Control Register

Base address: SSIE<sub>n</sub> = 0x4025\_D000 + 0x0100 × n (n = 0, 1)  
 SSIE<sub>n</sub>\_NS = 0x5025\_D000 + 0x0100 × n (n = 0, 1)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit field:	—	—	—	TDES[4:0]				—	—	—	RDFS[4:0]				—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Bit	Symbol	Function	R/W
4:0	RDFS[4:0]	RDF Setting Condition Select* <sup>1</sup> 0x00: SSIFRDR has one stage or more data size. 0x01: SSIFRDR has two stages or more data size. ⋮ 0x1E: SSIFRDR has thirty-one stages or more data size. 0x1F: SSIFRDR has thirty-two stages or more data size.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
12:8	TDES[4:0]	TDE Setting Condition Select* <sup>1</sup> 0x00: SSIFTDR has one stage or more free space. 0x01: SSIFTDR has two stages or more free space. ⋮ 0x1E: SSIFTDR has thirty-one stages or more free space. 0x1F: SSIFTDR has thirty-two stages or more free space.	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Writing to these bits while SSIE is in a communication state (SSISR.IIRQ = 0) is prohibited. If written, the operation performed immediately after writing is not guaranteed.

**RDFS[4:0] bits (RDF Setting Condition Select)**

The RDFS[4:0] bits set the setting condition of the receive data full flag (RDF).

**TDES[4:0] bits (TDE Setting Condition Select)**

The TDES[4:0] bits set the setting condition of the transmit data empty flag (TDE).

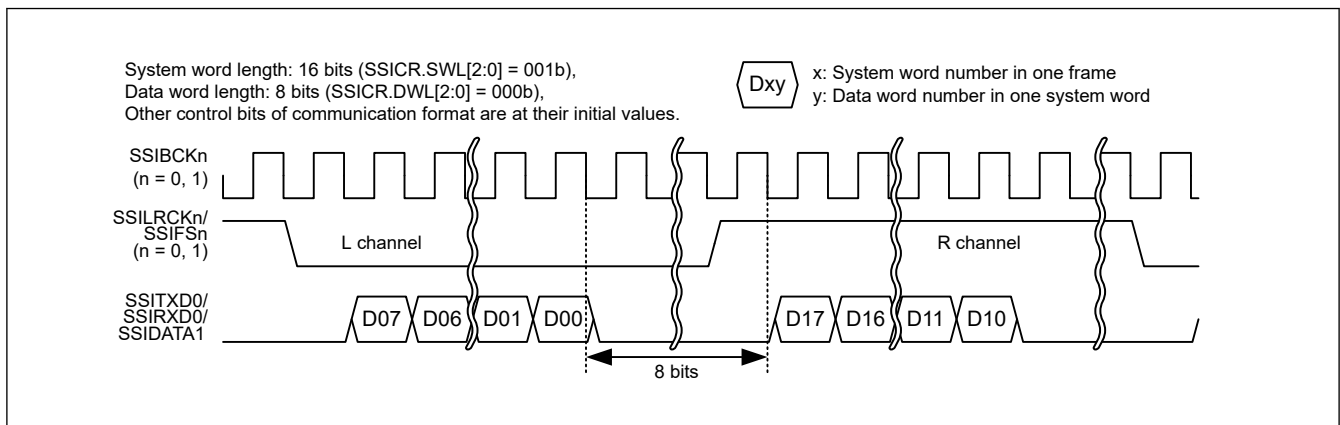
**37.3 Communication Formats**

SSIE supports three communication formats. Table 37.12 shows supported communication formats.

**Table 37.12 Supported communication formats**

Communication Format	SSIOFR.OMOD[1:0]
I <sup>2</sup> S format	00
TDM format	01
Monaural format	10

The following describes the serial data structure shared by communication formats. A serial data structure is defined by the system word length (set in SSICR.SWL[2:0]) and the data word length (set in SSICR.DWL[2:0]). If the data word length is shorter than the system word length, padding bits are transferred in the serial data. For details, see Figure 37.36.



**Figure 37.36 Example of padding bit transfer (I<sup>2</sup>S format: system word length > data word length)**

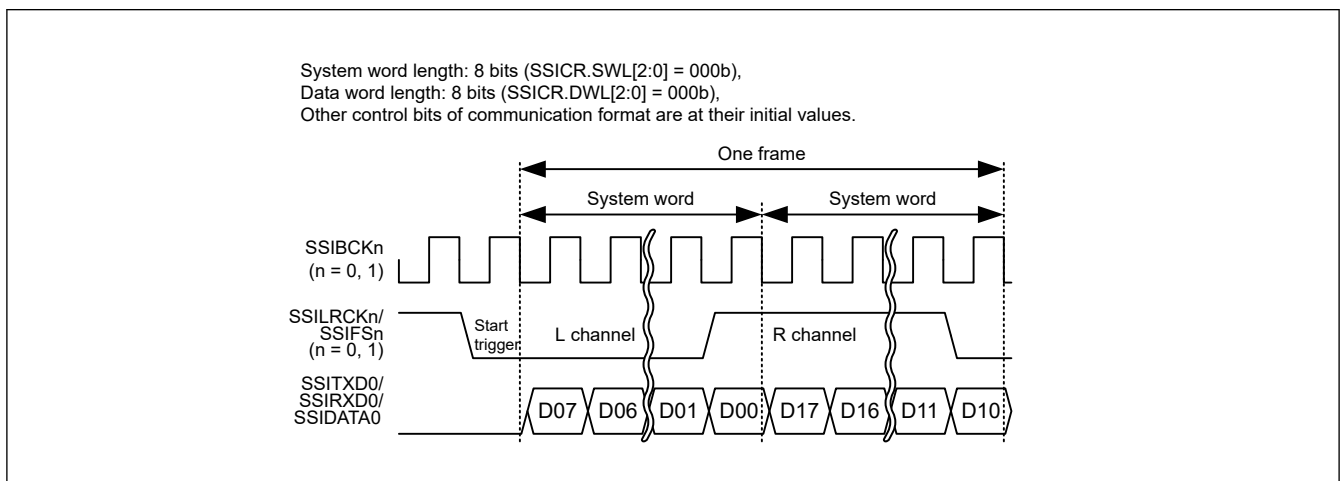
Table 37.13 lists the number of padding bits to be transferred with each combination of system word length (SSICR.SWL[2:0]) and data word length (SSICR.DWL[2:0]). “-” indicates that the setting is prohibited.

**Table 37.13** Number of padding bits

	SSICR.DWL[2:0]	000b	001b	010b	011b	100b	101b	110b	111b
SSICR.SWL[2:0]	System Word Length	8	16	18	20	22	24	32	Setting prohibited
000b	8	0	—	—	—	—	—	—	—
001b	16	8	0	—	—	—	—	—	—
010b	24	16	8	6	4	2	0	—	—
011b	32	24	16	14	12	10	8	0	—
100b	48	40	32	30	28	26	24	16	—
101b	64	56	48	46	44	42	40	32	—
110b	128	120	112	110	108	106	104	96	—
111b	256	248	240	238	236	234	232	224	—

### 37.3.1 I<sup>2</sup>S Format

The I<sup>2</sup>S format is a communication format used for connection with I<sup>2</sup>S-compatible serial devices. With this format setting (SSIOFR.OMOD[1:0] = 00b), one frame is configured with two system words, one for the channel L and the other for channel R. The SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) signals are at a low level for the channel L and at a high level for the channel R. Set the polarity of the signals with the SSICR.LRCKP bit. Figure 37.37 shows the I<sup>2</sup>S format without padding. See Figure 37.36 for the format with padding.



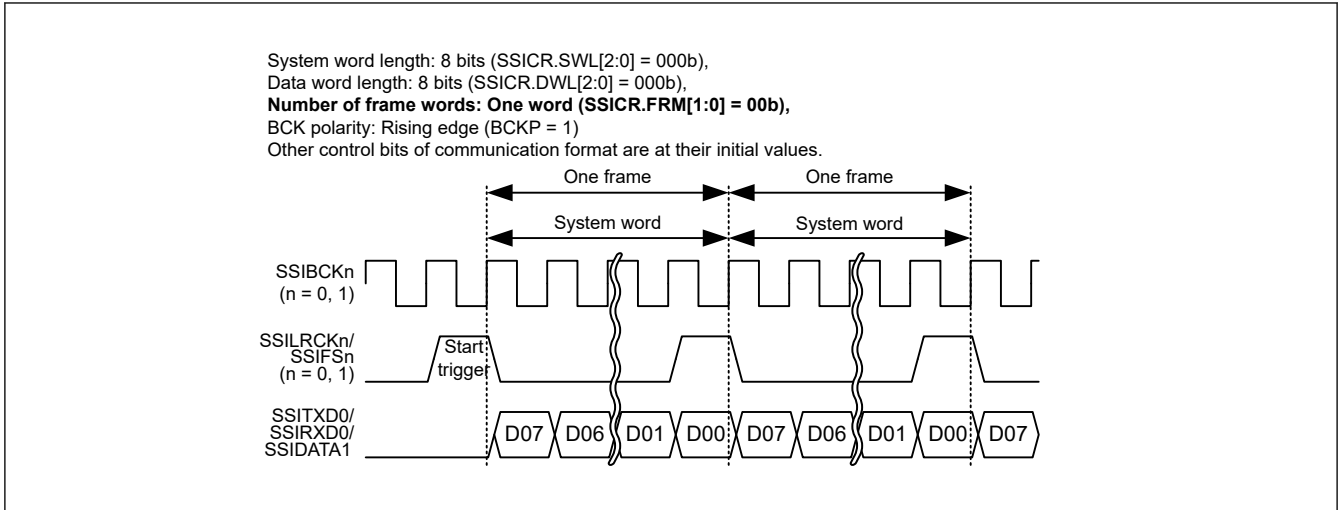
**Figure 37.37** I<sup>2</sup>S format (without padding: system word length = data word length)

For the state of external pins when SSIE is in the idle state, see section 37.5.1. Idle State.

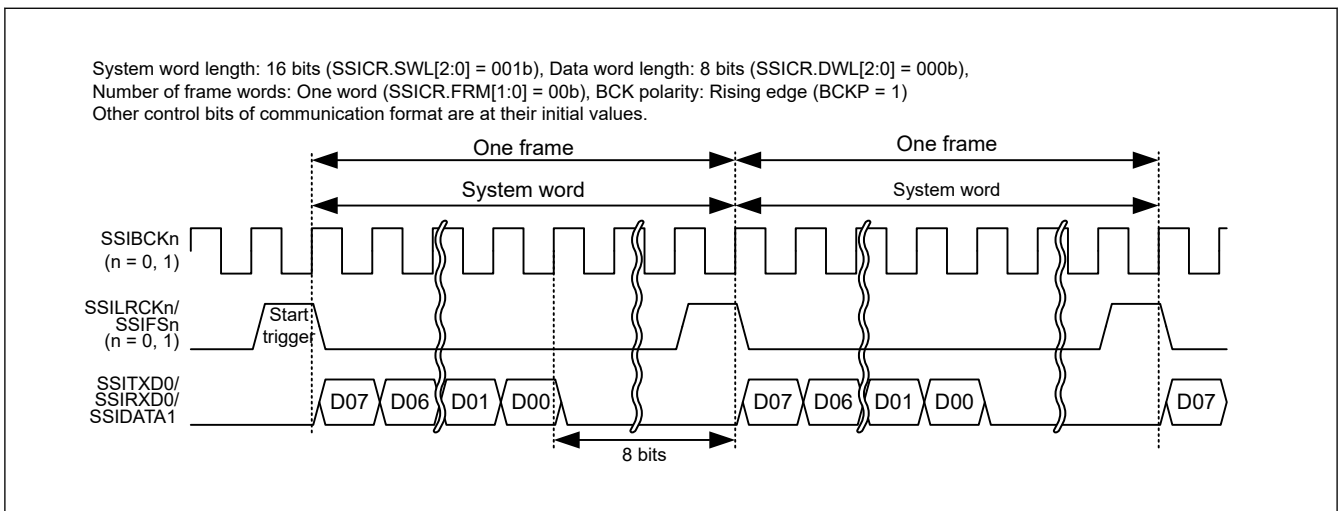
Note: SSIE has the SSILRCK<sub>n</sub>/SSIFS<sub>n</sub> (n = 0, 1) pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCK<sub>n</sub>/SSIFS<sub>n</sub> (n = 0, 1) pin only as a trigger to start communication.

### 37.3.2 Monaural Format

The monaural format is a communication format used for connection with monaural-compatible serial devices. When the monaural format is specified (SSIOFR.OMOD[1:0] = 10b) for use, one frame consists of one system word. Also, a rising edge of the SSILRCK<sub>n</sub>/SSIFS<sub>n</sub> (n = 0, 1) signal indicates a communication start trigger. Figure 37.38 and Figure 37.39 respectively show the monaural formats without and with padding.



**Figure 37.38 Short frame in monoaural format (without padding: system word length = data word length)**



**Figure 37.39 Short frame in monoaural format (with padding: system word length > data word length)**

The monoaural formats supported by SSIE consist of short frames and long frames. See [section 37.3.2.1. Short frame](#) and [section 37.3.2.2. Long frame](#) for the difference between these two frames.

For the state of external pins state when SSIE is in the idle state, see [section 37.5.1. Idle State](#).

Note: SSIE has the SSILRCKn/SSIFSs (n = 0, 1) pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCKn/SSIFSs (n = 0, 1) pin only as a trigger to start communication.

### 37.3.2.1 Short frame

When a short frame is used (SSICR.DEL = 0), the SSILRCKn/SSIFSs (n = 0, 1) signal indicating the start of serial data is set to high level only for 1 cycle of SSIBCKn (n = 0, 1). Data transfer starts at the falling edge of the signal.

### 37.3.2.2 Long frame

When a long frame is used (SSICR.DEL = 1), the SSILRCKn/SSIFSs (n = 0, 1) signal indicating the start of serial data is set to high level only for 2 cycles of SSIBCKn (n = 0, 1). See [Figure 37.40](#). Data transfer starts at the rising edge of the signal.

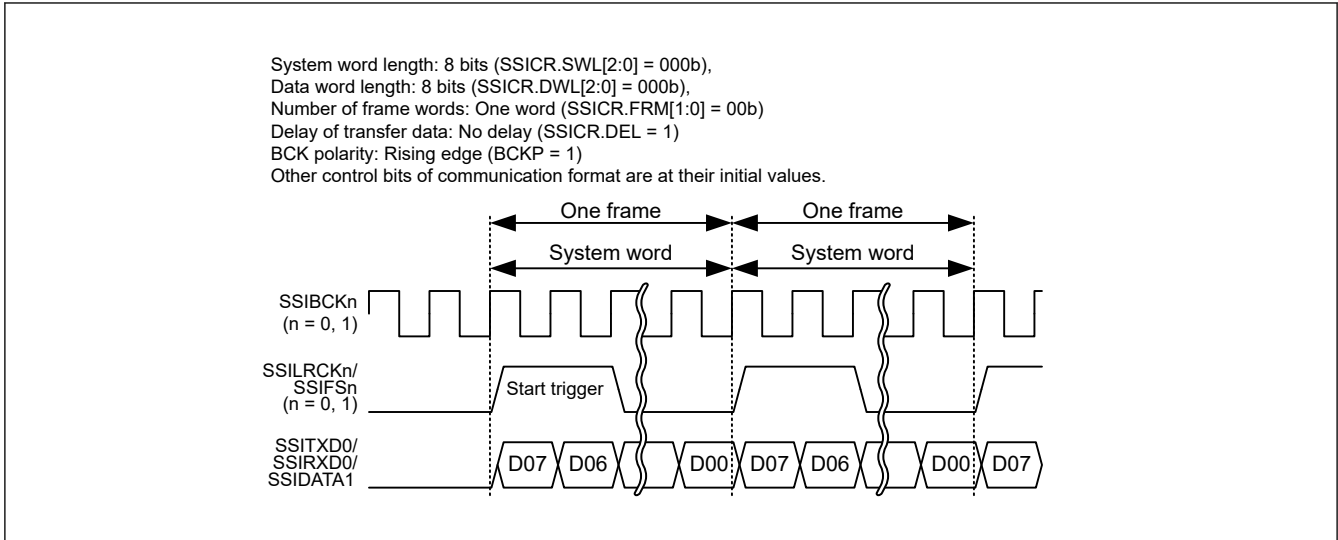


Figure 37.40 Long frame in monaural format (without padding)

### 37.3.3 TDM Format

The TDM format is a communication format used for connection with TDM-compatible multi-channel devices. With this format setting (SSIOFR.OMOD[1:0] = 01b), one frame is configured with four to eight system words set with the SSICR.FRM[1:0] bits. With this format, the SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) signal is at a high level for the first one system word and at a low level for the rest. The pulse generated on the SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) signal is defined as the SYNC pulse and its rising edge means a start of one frame. Figure 37.41 and Figure 37.42 respectively show the TDM formats without and with padding.

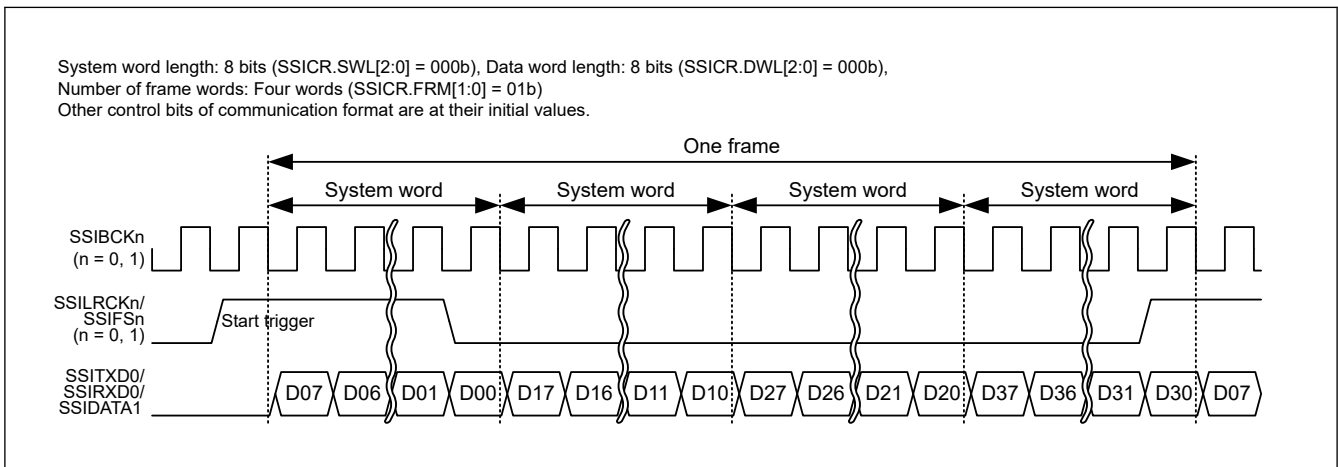


Figure 37.41 TDM format (without padding: system word length = data word length)

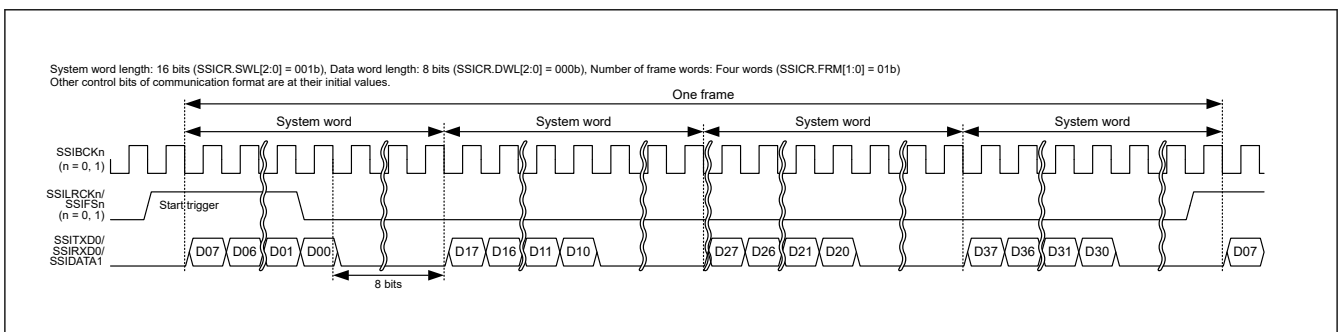


Figure 37.42 TDM format (with padding: system word length > data word length)

For the state of external pins when SSIE is in the idle state, see [section 37.5.1. Idle State](#).



Note: SSIE has the SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) pin only as a trigger to start communication.

## 37.4 Communication Modes

SSIE supports the following communication modes. Table 37.15 lists the control bits that are not available with each communication mode. See section 37.4.1. Slave-mode Communication to section 37.4.5. Transmission and Reception for details of these communication modes.

**Table 37.14 Communication modes**

Communication Mode	SSICR.MST Bit	SSICR.REN Bit	SSICR.TEN Bit
Slave-mode transmission	0	0	1
Slave-mode reception	0	1	0
Slave-mode transmission and reception	0	1	1
Master-mode transmission	1	0	1
Master-mode reception	1	1	0
Master-mode transmission and reception	1	1	1

**Table 37.15 Control bits that cannot be used in each communication mode**

Control Bit	Slave-mode Reception	Slave-mode Transmission	Slave-mode Transmission and Reception	Master-mode Reception	Master-mode Transmission	Master-mode Transmission and Reception
SSICR.CKS	Invalid	Invalid	Invalid	Available	Available	Available
SSICR.CKDV	Invalid	Invalid	Invalid	Available	Available	Available
SSICR.MUEN	Invalid	Available	Available	Invalid	Available	Available
SSICR.TEN	Invalid	Available	Available	Invalid	Available	Available
SSICR.REN	Available	Invalid	Available	Available	Invalid	Available
SSIFCR.AUCKEN	Invalid	Invalid	Invalid	Available	Available	Available
SSIFCR.TIE	Invalid	Available	Available	Invalid	Available	Available
SSIFCR.RIE	Available	Invalid	Available	Available	Invalid	Available
SSIFCR.TFRST	Invalid	Available	Available	Invalid	Available	Available
SSIFCR.RFRST	Available	Invalid	Available	Available	Invalid	Available
SSIOFR.BCKASTP	Invalid	Invalid	Invalid	Available	Available	Available
SSIOFR.LRCONT	Invalid	Invalid	Invalid	Available	Available	Available
SSIOFR.OMOD	Available	Available	Available	Available	Available	Available
SSISCR.TDES	Invalid	Available	Available	Invalid	Available	Available
SSISCR.RDFS	Available	Invalid	Available	Available	Invalid	Available

“Invalid” means it has no effect on operation. Writing is possible.

### 37.4.1 Slave-mode Communication

SSIE operates in slave mode with SSICR.MST = 0. The SSIBCK<sub>n</sub> (n = 0, 1) and SSILRCK<sub>n</sub>/SSIFS<sub>n</sub> (n = 0, 1) signals to be used for serial-data communication must be supplied from an external device. If these signals do not match the communication format set for SSIE, operation is not guaranteed.

### 37.4.2 Master-mode Communication

SSIE operates in master mode with `SSICR.MST = 1`. The `SSIBCKn` ( $n = 0, 1$ ) and `SSILRCKn/SSIFSn` ( $n = 0, 1$ ) signals to be used for serial data communication must be internally generated from the audio clock. These signals use the format according to the setting of SSIE. If the communication format the slave device uses does not match the communication format set for SSIE, the operation is unpredictable.

### 37.4.3 Transmission

SSIE transmits serial data to the other-party device when the `SSICR.TEN` bit is 1 and the `SSICR.REN` bit is 0. If the communication format the other-party device uses does not match the communication format set for SSIE, the operation is unpredictable.

### 37.4.4 Reception

SSIE receives serial data from the other-party device when the `SSICR.TEN` bit is 0 and the `SSICR.REN` bit is 1. If the communication format the other-party device uses does not match the communication format set for SSIE, the operation is unpredictable.

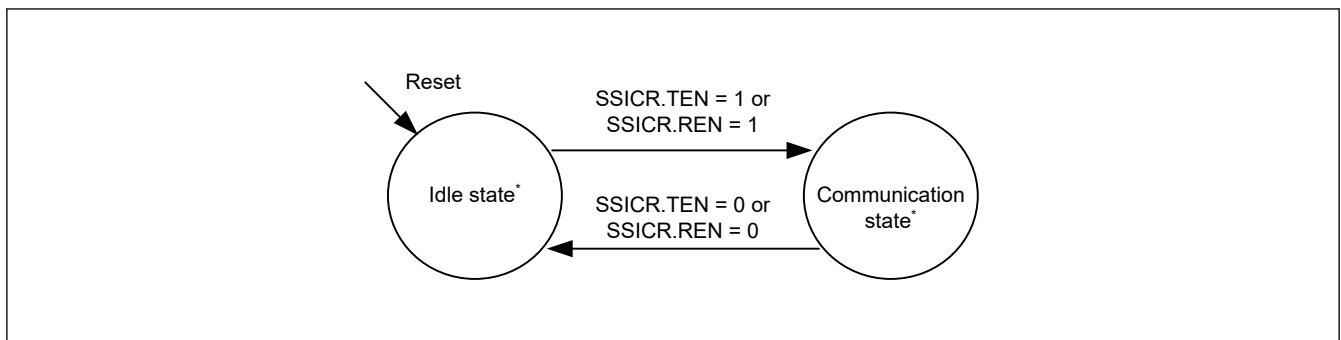
### 37.4.5 Transmission and Reception

SSIE transmits and receives serial data to and from the other-party device when the `SSICR.TEN` bit is 1 and the `SSICR.REN` bit is 1. If the communication format the other-party device uses does not match the communication format set for SSIE, the operation is unpredictable.

## 37.5 Operation

SSIE has the following two main operation states [Figure 37.43](#) shows SSIE state transition.

- Idle state (`SSISR.IIRQ = 1`)
- Communication state (`SSISR.IIRQ = 0`).



**Figure 37.43** SSIE state transition

Note: See [section 37.6.1. Start Communication](#) for details of the idle state.

See [section 37.6.2. Transmission](#) for details of the communication state.

### 37.5.1 Idle State

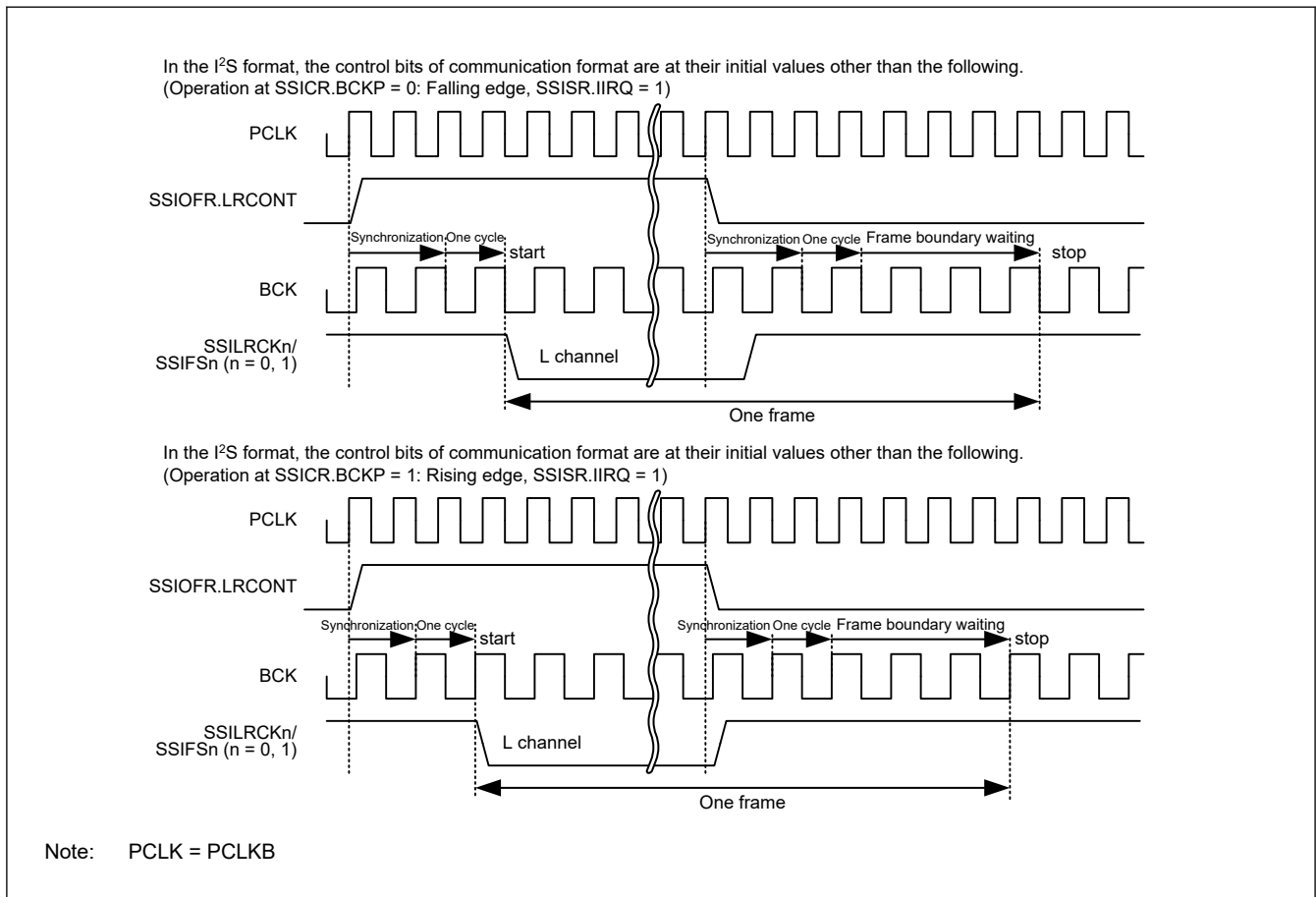
In this state, communication of SSIE is halted. If, however, the `SSICR.MST` bit is 1, output of the BCK and LR clock/frame synchronization signals to external pins can be controlled according to the settings of `SSIOFR.BCKASTP` and `SSIOFR.LRCONT` bits. This function is common to all formats. For details, see [Table 37.16](#).

**Table 37.16** Output from external pins in the idle state (1 of 2)

SSICR.MST	SSIOFR.BCKASTP	SSIOFR.LRCONT	Output from Pins		
			SSIBCK <sub>n</sub> (n = 0, 1)	SSILRCK <sub>n</sub> /SSIFS <sub>n</sub> (n = 0, 1)	SSITXD0/SSIDATA1
0	—	—	Stop	Stop	Stop

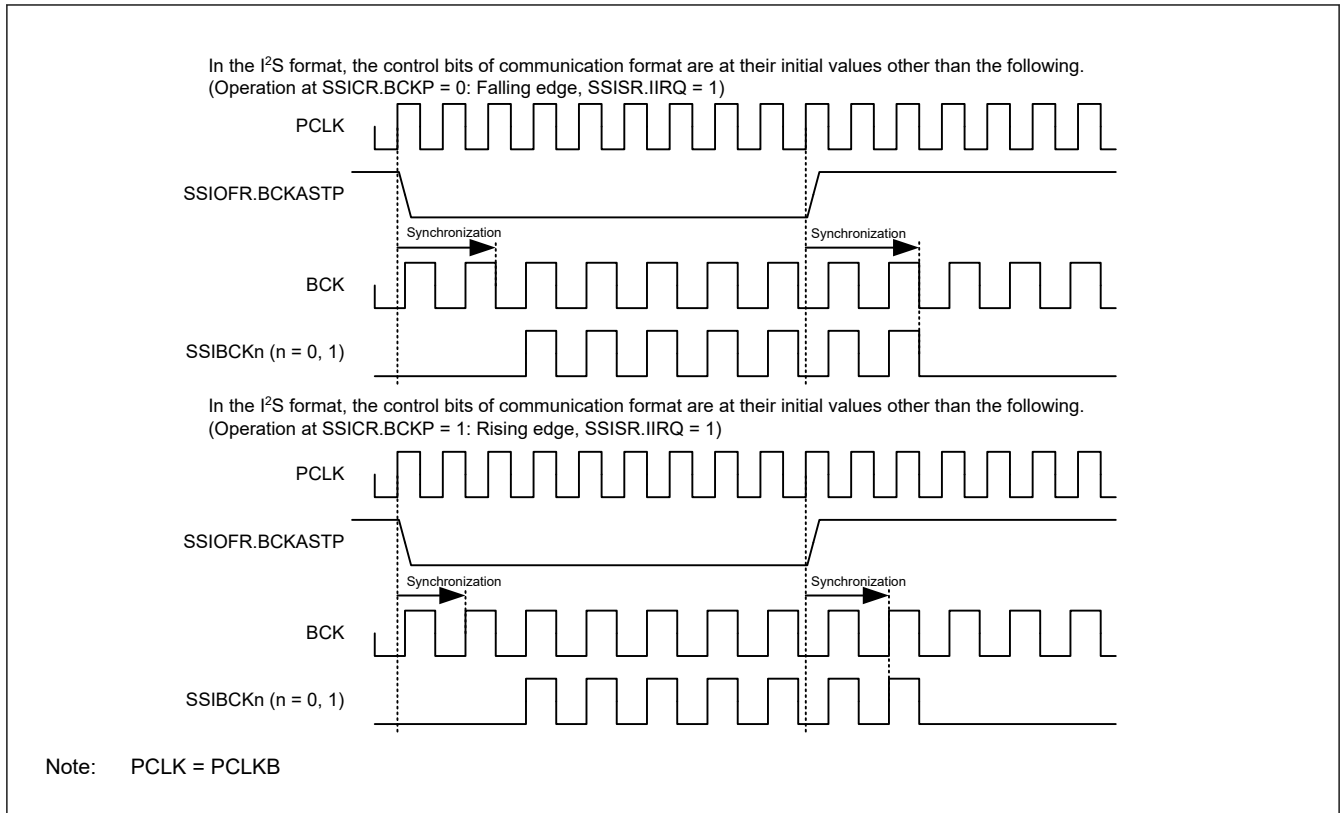
**Table 37.16 Output from external pins in the idle state (2 of 2)**

SSICR.MST	SSIOFR.BCKASTP	SSIOFR.LRCONT	Output from Pins		
			SSIBCKn (n = 0, 1)	SSILRCKn/SSIFS <sub>n</sub> (n = 0, 1)	SSITXD0/SSIDATA1
1	0	0	Supply	Stop	Stop
1	0	1	Supply	Supply	Stop
1	1	0	Stop	Stop	Stop
1	1	1	Stop	Supply	Stop



**Figure 37.44 Example of disabling LR clock/frame synchronization continuation by SSIOPFR.LRCONT**

Note: To stop the output to the SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) pin with SSIOPFR.LRCONT when SSIE is in the idle state in master-mode communication (SSICR.MST = 1), note the following: The output stops when the value of the SSIOPFR.LRCONT bit is changed from 1 to 0. Make sure that the other-party device is not affected.



**Figure 37.45 Example of stopping SSIBCKn (n = 0, 1) with SSIOFR.BCKASTP**

Note: To stop the output to the SSIBCKn (n = 0, 1) pin with SSIOFR.BCKASTP in master-mode communication (SSICR.MST = 1) and while SSIE is in the idle state, note the following: The output stops when the value of the SSIOFR.BCKASTP bit is changed from 0 to 1. So, make sure that the other-party device is not affected.

### 37.5.2 Communication States

In this state, SSIE is during communication. [Figure 37.46](#) shows transitions of communication states and [Table 37.17](#) lists the conditions for transition. If the transition condition is not satisfied, the state does not transit.

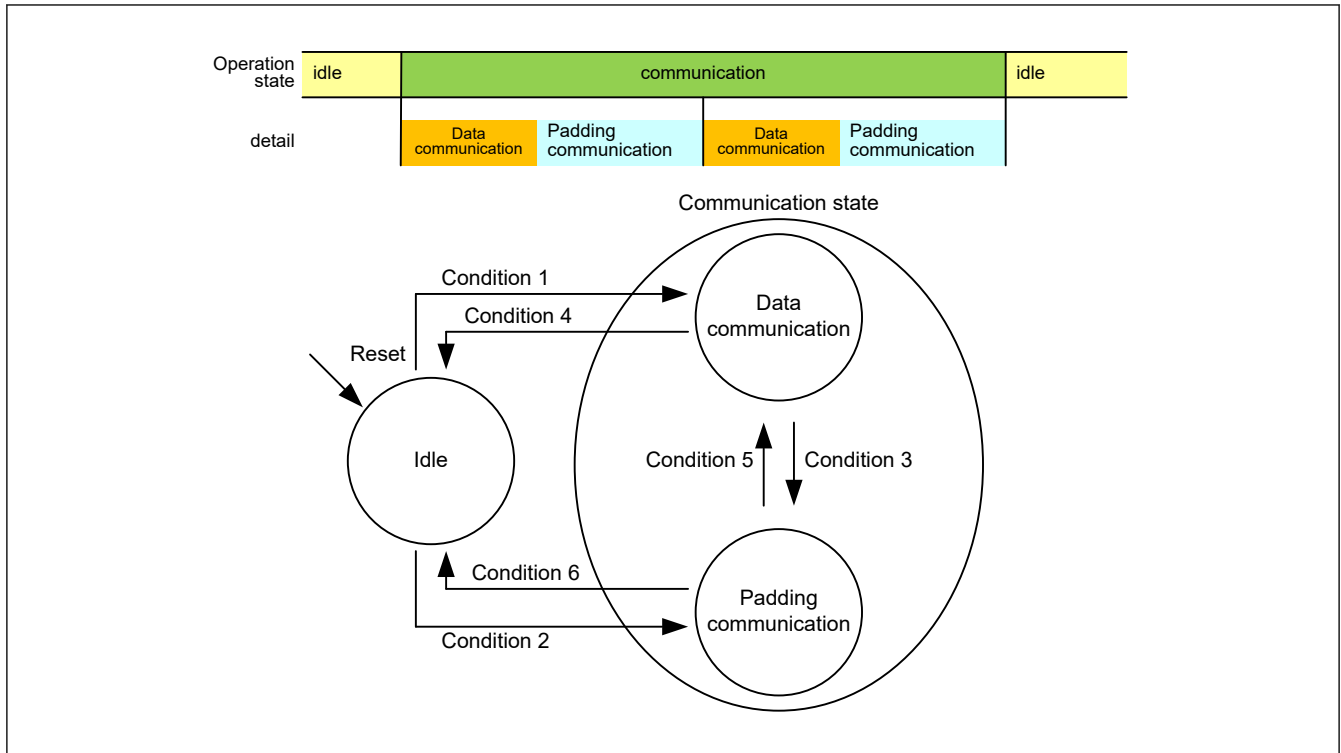


Figure 37.46 Communication state transition

Table 37.17 Condition for communication state transition

Condition Number	Condition for Transition
1	Writing SSICR.TEN = 1 or SSICR.REN = 1 while SSICR.SDTA = 0 or in the setting without padding bits.
2	Writing SSICR.TEN = 1 or SSICR.REN = 1 while SSICR.SDTA = 1 and in the setting with padding bits.
3	The following three conditions are all met: <ul style="list-style-type: none"> <li>• SSICR.TEN = 1 or SSICR.REN = 1</li> <li>• In the setting with padding bits</li> <li>• The last bit of the data words has been transferred.</li> </ul>
4	Both the following two conditions are met: <ul style="list-style-type: none"> <li>• SSICR.SDTA = 1 or without padding bits</li> <li>• While SSICR.TEN = 0 and SSICR.REN = 0, the last bit of the data words in a frame has been transferred.</li> </ul>
5	Transfer of the last padding bit is completed while SSICR.TEN = 1 or SSICR.REN = 1
6	Both the following two conditions are met: <ul style="list-style-type: none"> <li>• SSICR.SDTA = 0 and with padding bits</li> <li>• While SSICR.TEN = 0 and SSICR.REN = 0, the last padding bit has been transferred.</li> </ul>

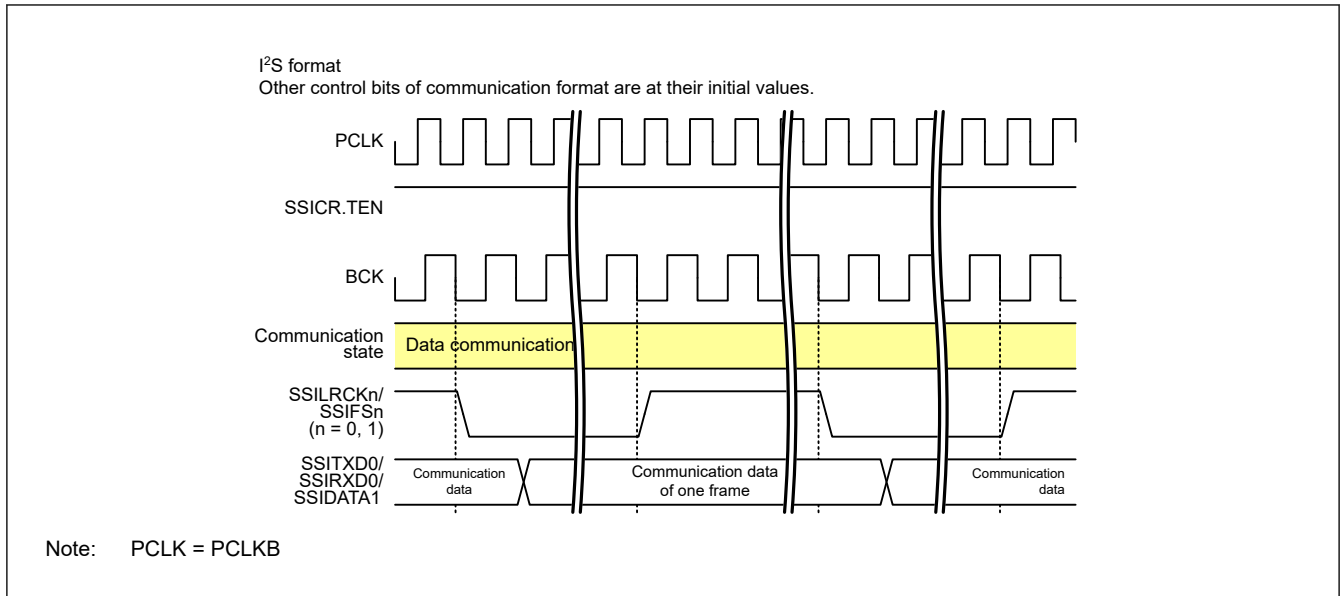
See Table 37.13 for the setting with/without padding bits.

### 37.5.2.1 Data communication state

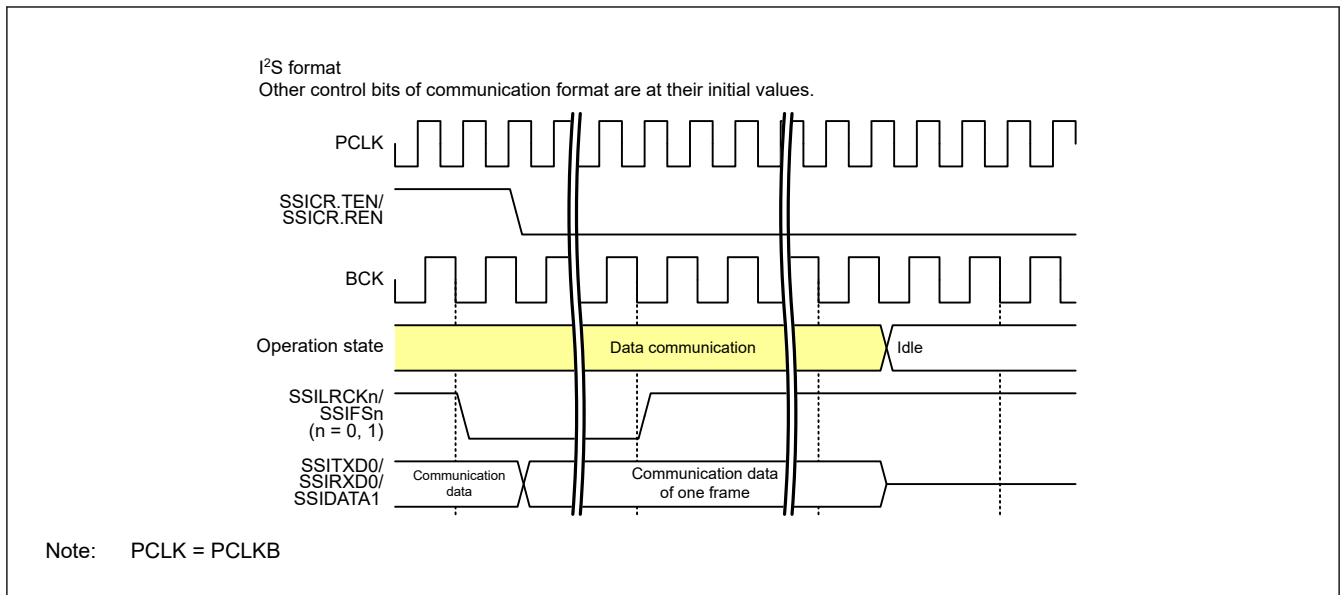
In this state, SSIE is during communication. Data of the data word length set with the SSICR.DWL[2:0] bits is transmitted, received, or transmitted and received.

- State Transition in the Setting without Padding Bits

During communication (SSISR.IIRQ = 0), SSIE is during data communication for all the time. By disabling transmission and reception (SSICR.TEN = 0, SSICR.REN = 0), SSIE transits to the idle state. For details, see Figure 37.47 and Figure 37.48.



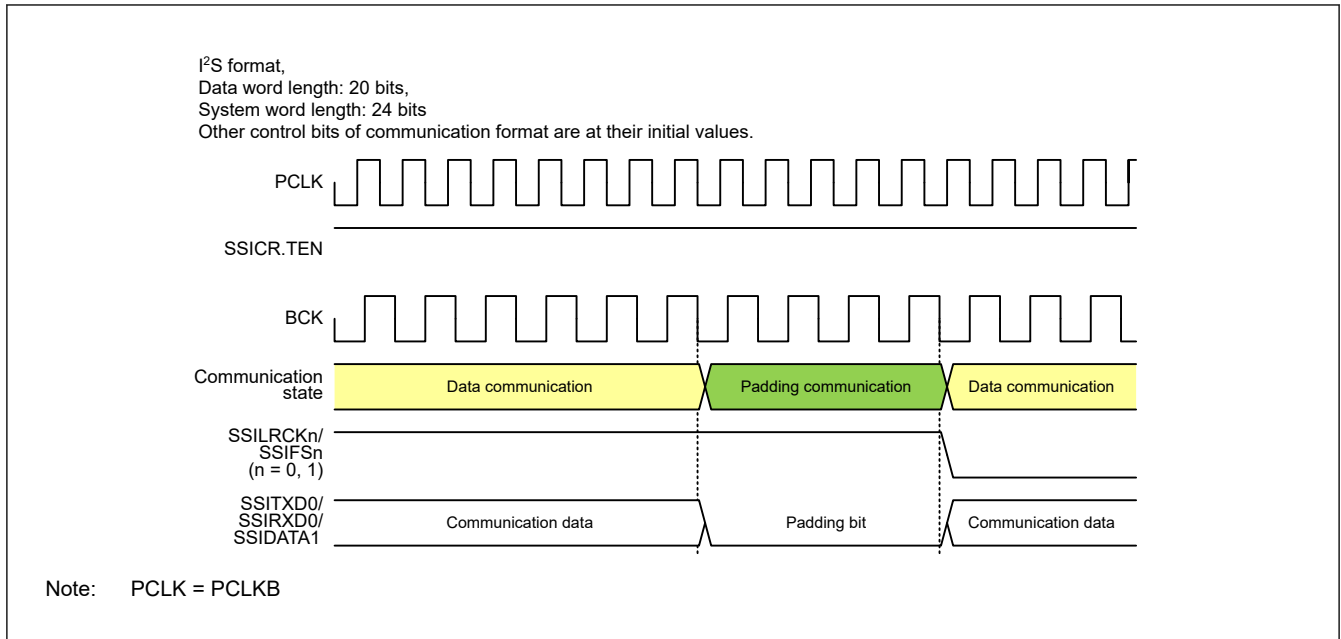
**Figure 37.47 Continuation of the data communication**



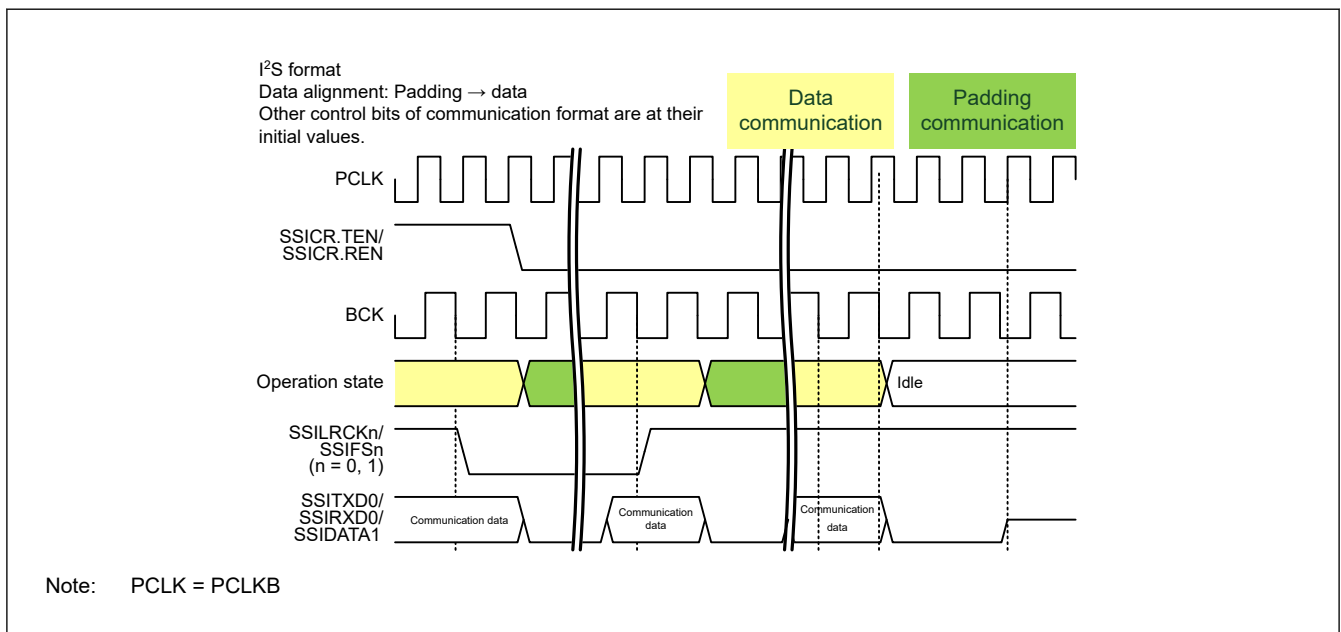
**Figure 37.48 Halt from the data communication (without padding bits)**

- State Transition in the Setting with Padding Bits

When SSIE ends transfer of the last bit of a data word during communication (SSISR.IIRQ = 0), SSIE transitions from the data communication state to the padding communication state in [Figure 37.49](#). Except in the status with SSICR.SDTA = 1 and transmission and reception disabled (SSICR.TEN = 0 and SSICR.REN = 0), SSIE transitions from the data communication state to the idle state when it stops communication in [Figure 37.50](#).



**Figure 37.49 Transition from data communication to padding communication**



**Figure 37.50 Halt from data communication (with padding bits)**

### 37.5.2.2 Padding communication

In this state, SSIE is during communication. The padding bits set with the SSICR.SWL[2:0] bits and SSICR.DWL[2:0] bits are transmitted, received, or transmitted and received.

- State Transition in the Setting with Padding Bits

When SSIE ends transfer of the last padding bit during communication (SSISR.IIRQ = 0), SSIE transitions to the data communication state in [Figure 37.49](#). If SSIE is in the status with SSICR.SDTA = 0 and transmission and reception disabled (SSICR.TEN = 0 and SSICR.REN = 0), SSIE transitions from the padding communication state to the idle state when it stops communication in [Figure 37.51](#).

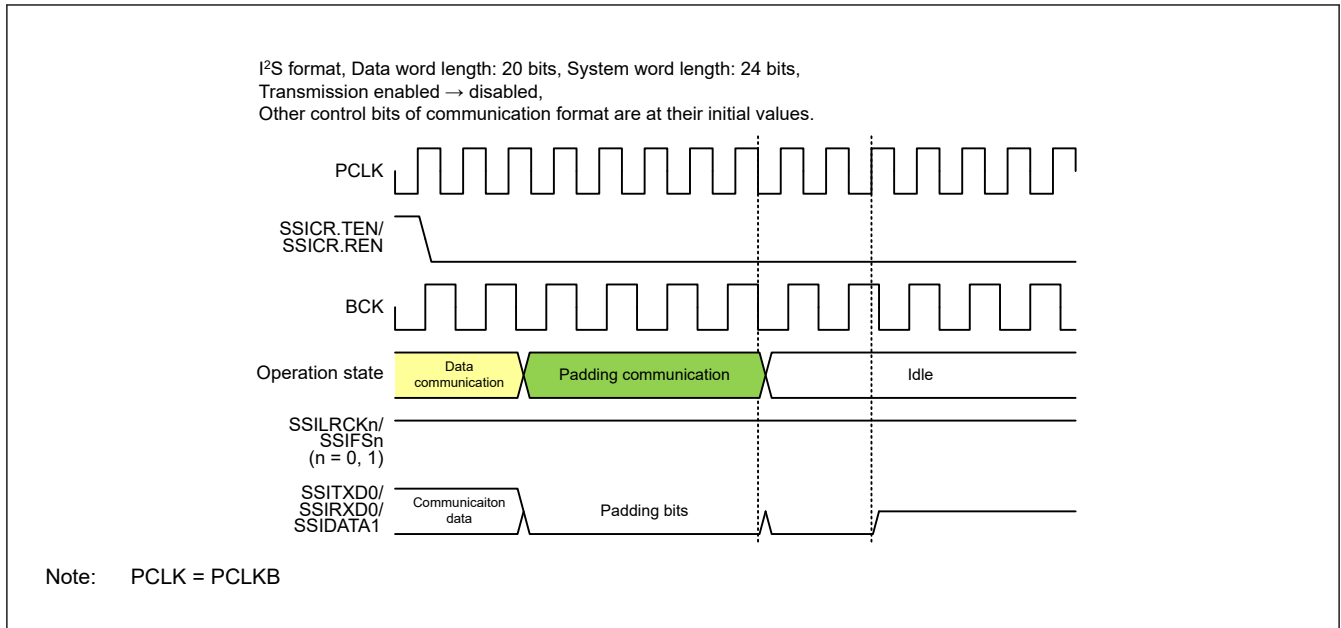


Figure 37.51 Halt from the padding communication

### 37.6 Communication Operation

Figure 37.52 shows the communication flow of SSIE.

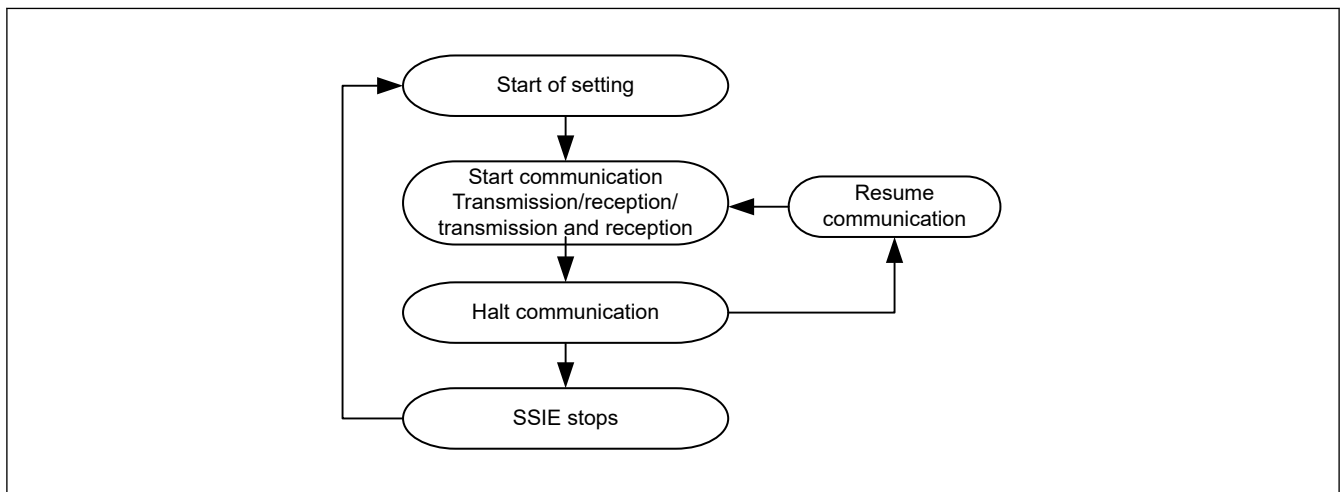


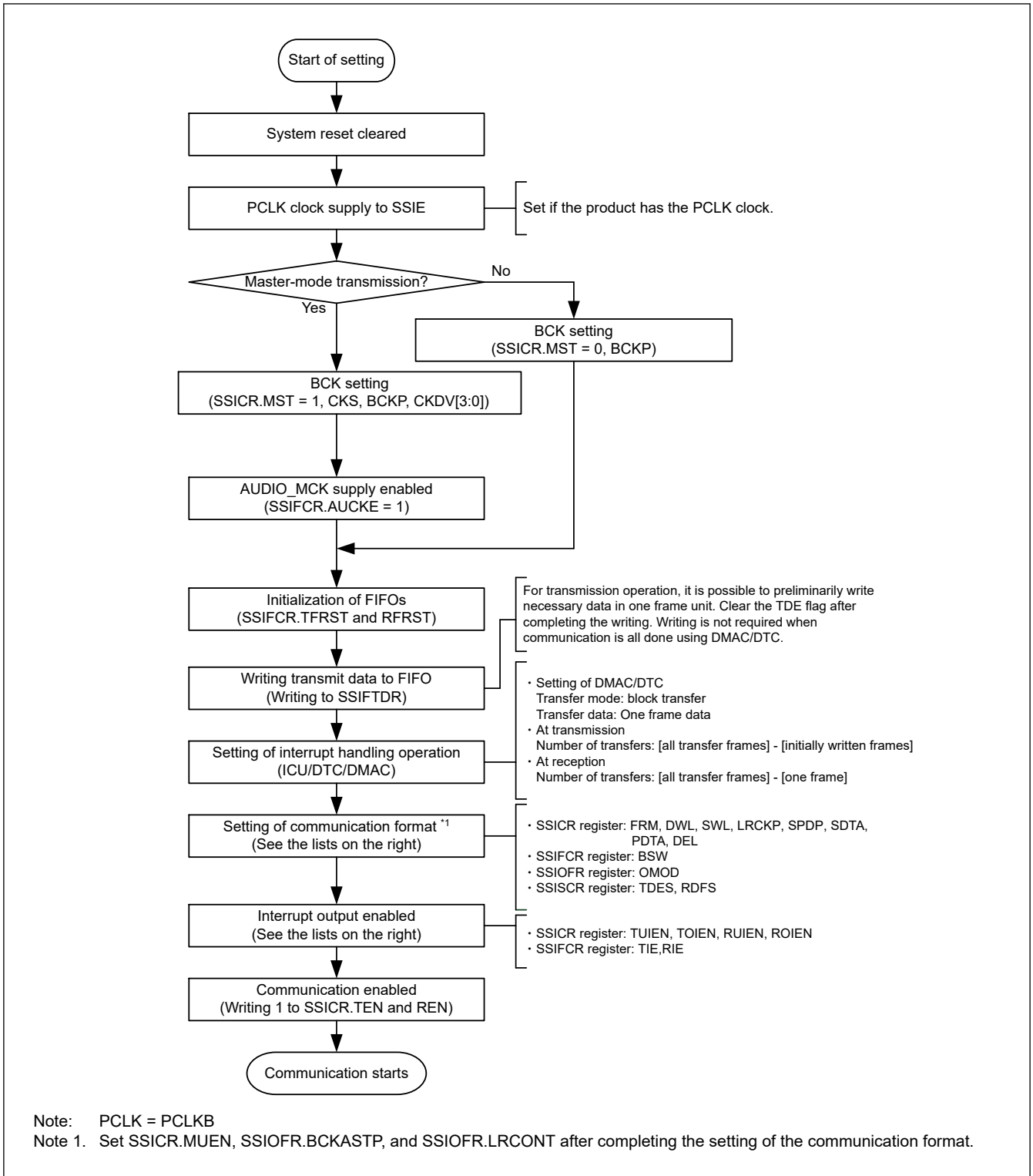
Figure 37.52 SSIE communication operation

The procedure of each operation is described from [section 37.6.1. Start Communication](#) to [section 37.6.7. Resume Communication](#).

#### 37.6.1 Start Communication

This section describes how to start communication of SSIE. [Figure 37.53](#) shows the procedure to start communication. Be sure to follow the procedure. See [section 37.6.2. Transmission](#) for transmission operation and [section 37.6.3. Reception](#) for reception operation.





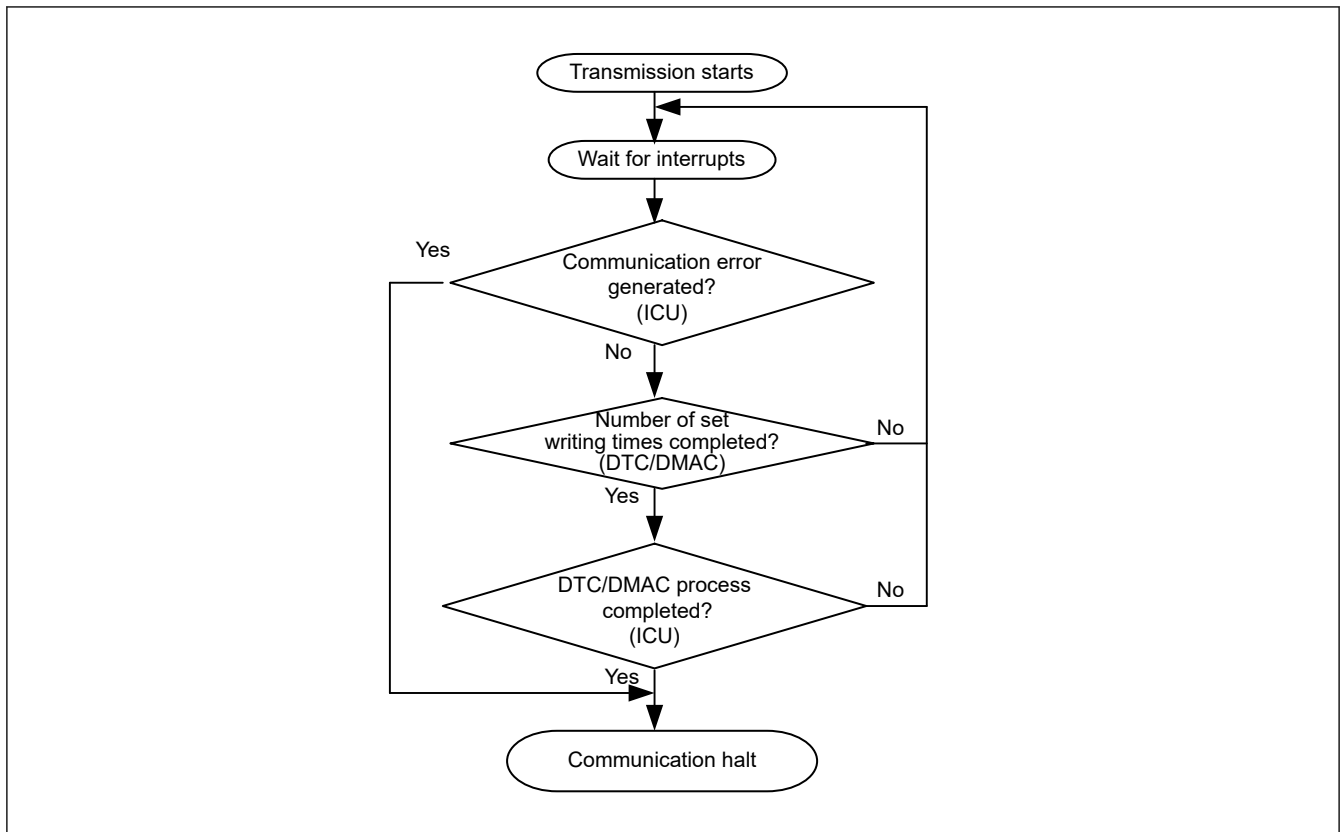
**Figure 37.53 Procedure to start communication (CPU operation procedure)**

SSIE can perform continuous communication based on interrupts by the DTC/DMAC. For transmission, write 1 to SSIFCR.TIE, SSICR.TUIEN, and SSICR.TOIEN. For reception, write 1 to SSIFCR.RIE, SSICR.RUIEN, and SSICR.ROIEN.

### 37.6.2 Transmission

The transmission procedure in [Figure 37.54](#) must be followed throughout a transmission operation.

After transmission is enabled (SSICR.TEN = 1 and SSICR.REN = 0), SSIE starts transmission when a start trigger is generated by SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) with the serial data for at least a frame contained in the transmit FIFO data register (SSIFTDR). SSIE outputs a transmit data empty interrupt to the DTC/DMAC according to the TDE setting condition (SSISCR.TDES) and the status of transmit data empty interrupt enable (SSIFCR.TIE) bit specified in the communication start procedure. This interrupt requests writing to the transmit FIFO data register (SSIFTDR). In the communication start procedure, specify writing to the transmit FIFO data register (SSIFTDR) as the DTC/DMAC operation in response to the transmit data empty interrupt. With this setting, SSIE can continuously transmit data not through the CPU. The transmit data empty interrupt is generated when the free space size of transmit FIFO data register reaches the value set in SSISCR.TDES. The number of times of writing must be specified in accordance with the free space size of the transmit FIFO data register indicated by the transmit data empty interrupt. If an error occurs, perform the error-handling procedure as instructed in the communication stop procedure.



**Figure 37.54** Transmission procedure

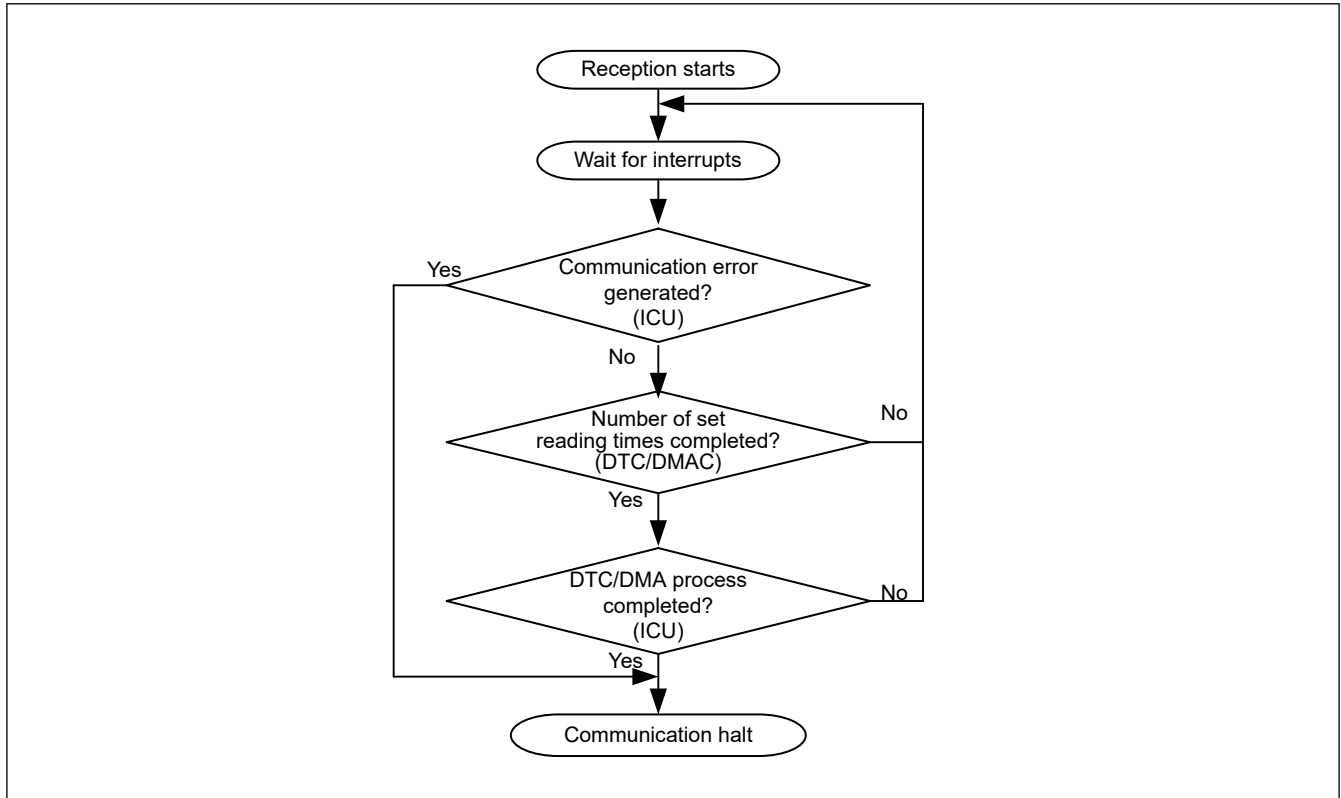
**Note:** The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.TDE to write data to SSIFTDR. The number of times of writing data to SSIFTDR by detecting the value 1 of SSIFSR.TDE must be in accordance with the free space size of the transmit FIFO data register specified by SSISCR.TDES. After as much transmit data as the free space size is written to SSIFTDR, the SSIFSR.TDE flag must be cleared. Continuous transmission is enabled by repeating data writing. If the SSIFSR.TDE flag is not cleared, the flag is not cleared automatically.

### 37.6.3 Reception

The reception procedure in [Figure 37.55](#) must be followed throughout a reception operation.

After reception is enabled (SSICR.TEN = 0 and SSICR.REN = 1), SSIE starts reception when a start trigger is generated by SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1). SSIE outputs a receive data full interrupt to the DTC/DMAC according to the RDF setting condition (SSISCR.RDFS) and the status of receive data full interrupt enable (SSIFCR.RIE) bit specified in the communication start procedure. This interrupt requests data reading from the receive FIFO data register (SSIFRDR). In the communication start procedure, specify reading from the receive FIFO data register (SSIFRDR) as the DTC/DMAC operation in response to the receive data full interrupt. With this setting, SSIE can continuously read data not through the CPU. The receive data full interrupt is generated when data as much as the capacity of receive FIFO data register has been stored. The number of times of reading must be specified in accordance with the data size of the receive FIFO data register

indicated by the receive data full interrupt. If an error occurs, perform the error-handling procedure as instructed in the communication stop procedure.



**Figure 37.55 Reception procedure**

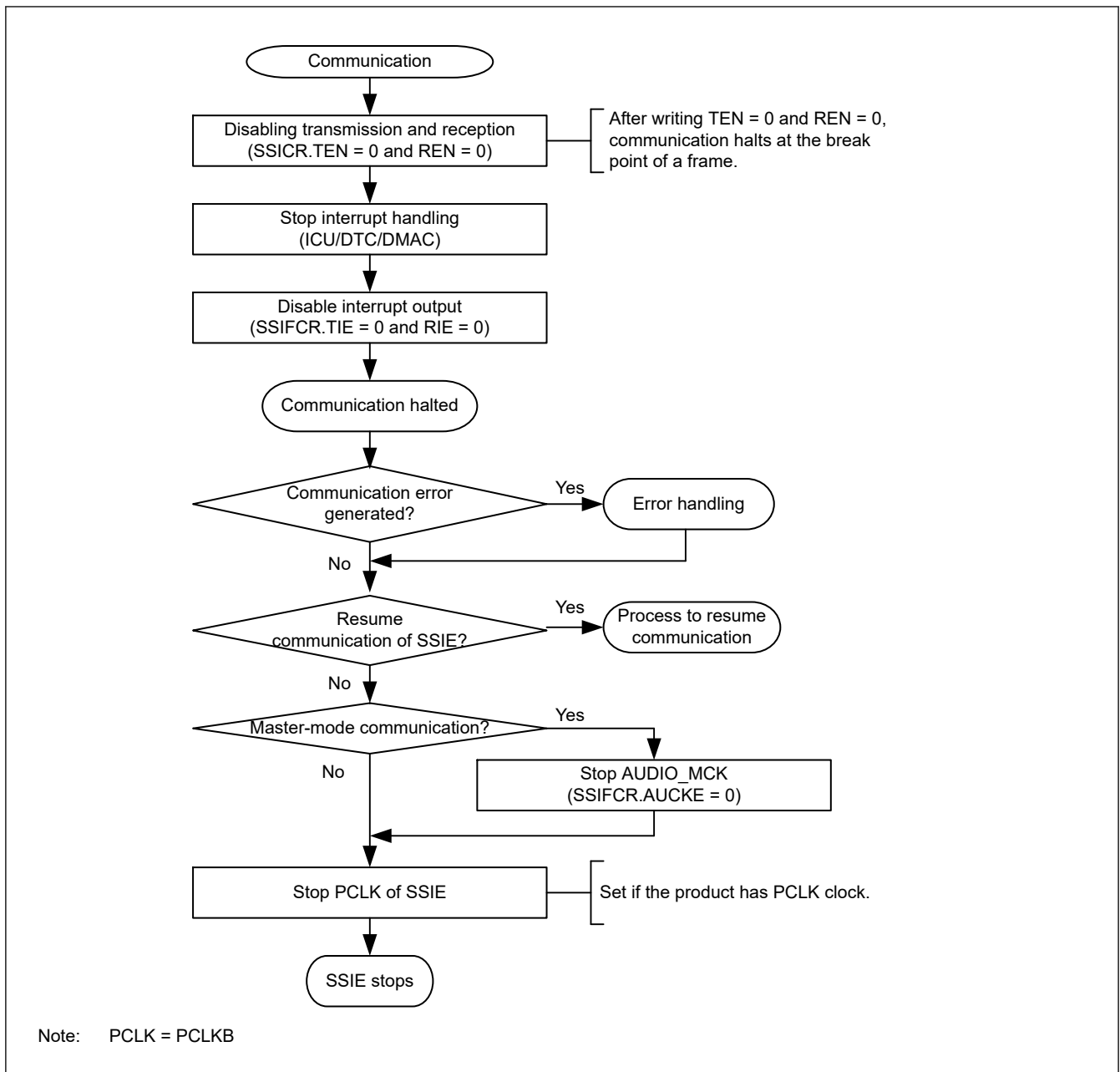
**Note:** The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.RDF to read data from SSIFRDR. The number of times of reading data from SSIFRDR by detecting the value 1 of SSIFSR.RDF must be in accordance with the receive data storage capacity of the receive FIFO data register specified by SSISCR.RDFS. After received data is read from SSIFRDR, the SSIFSR.RDF flag must be cleared. Continuous reception is enabled by repeating data reading. If the SSIFSR.RDF flag is not cleared, the flag is not cleared automatically.

### 37.6.4 Transmission and Reception

After transmission and reception are enabled (SSICR.TEN = 1 and SSICR.REN = 1), SSIE starts transmission and reception when a start trigger is generated by SSILRCKn/SSIFSn (n = 0, 1) with the serial data for at least a frame contained in the transmit FIFO data register (SSIFTDR). SSIE can continuously transmit and receive data by performing the procedures described in [section 37.6.2. Transmission](#) and [section 37.6.3. Reception](#), respectively. For how to stop transmission and reception, see [section 37.6.5. Halt Communication](#).

### 37.6.5 Halt Communication

This section describes how to halt communication of SSIE. [Figure 37.56](#) shows the procedure to halt communication. Be sure to follow the procedure.



**Figure 37.56 Procedure to halt communication (CPU operation procedure)**

To halt the communication of SSIE, supply of the following clocks are required until the SSISR.IIRQ bit indicates an idle state.

- Input clock from the SSIBCKn (n = 0, 1) pin when SSICR.MST = 0
  - AUDIO\_MCK when SSICR.MST = 1
- To resume communication of SSIE in the previous setting, see [section 37.6.7. Resume Communication](#).

Note: When communication of SSIE is halted according to the procedure to halt communication in [Figure 37.56](#), resume communication according to the procedure to resume communication in [Figure 37.58](#).

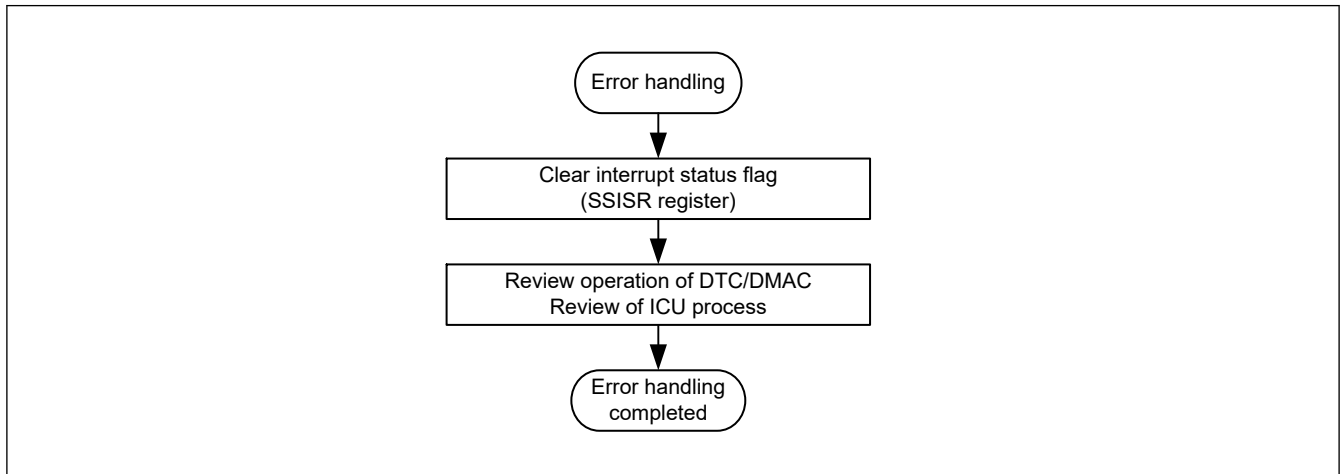
### 37.6.6 Error Handling

SSIE has the following four errors.

- Transmit underflow error
- Transmit overflow error
- Receive underflow error

- Receive overflow error.

When an underflow error or overflow error is generated, SSIE need to be restarted. Follow the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#).



**Figure 37.57 Error-handling procedure**

Four error operations are described as follows. When the interrupt output enable bit of the SSICR register is enabled and error flags are set, an error interrupt is generated. See [section 37.2.2. SSISR : Status Register](#) for the setting conditions of error flags.

#### (1) Transmit Underflow Error

If a transmit underflow error occurs, review the number of times of writing data to the transmit FIFO data register (SSIFTDR) in response to a transmit data empty interrupt. After a transmit underflow error occurs, SSIE outputs 0s as data. To normally output the serial data written to the transmit FIFO data register (SSIFTDR) to the SSITXD0/SSIDATA1 pin, follow the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#). After this error occurs, serial data is consumed as usual. If you resume communication, write the serial data from the beginning.

#### (2) Transmit Overflow Error

If a transmit overflow error occurs, review the number of times of writing data to the transmit FIFO data register (SSIFTDR) in response to transmit data empty interrupts. The serial data written to the transmit FIFO data register (SSIFTDR) that caused the transmit overflow error becomes invalid. This error can occur regardless of whether a transmission operation is being done. To recover from the error, follow the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#). When you resume communication, deal with the invalid serial data appropriately.

#### (3) Receive Underflow Error

If a receive underflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The values read from the receive FIFO data register (SSIFRDR) that caused the receive underflow error are undefined. This error can occur regardless of whether a reception operation is being done. To recover from the error, follow the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#).

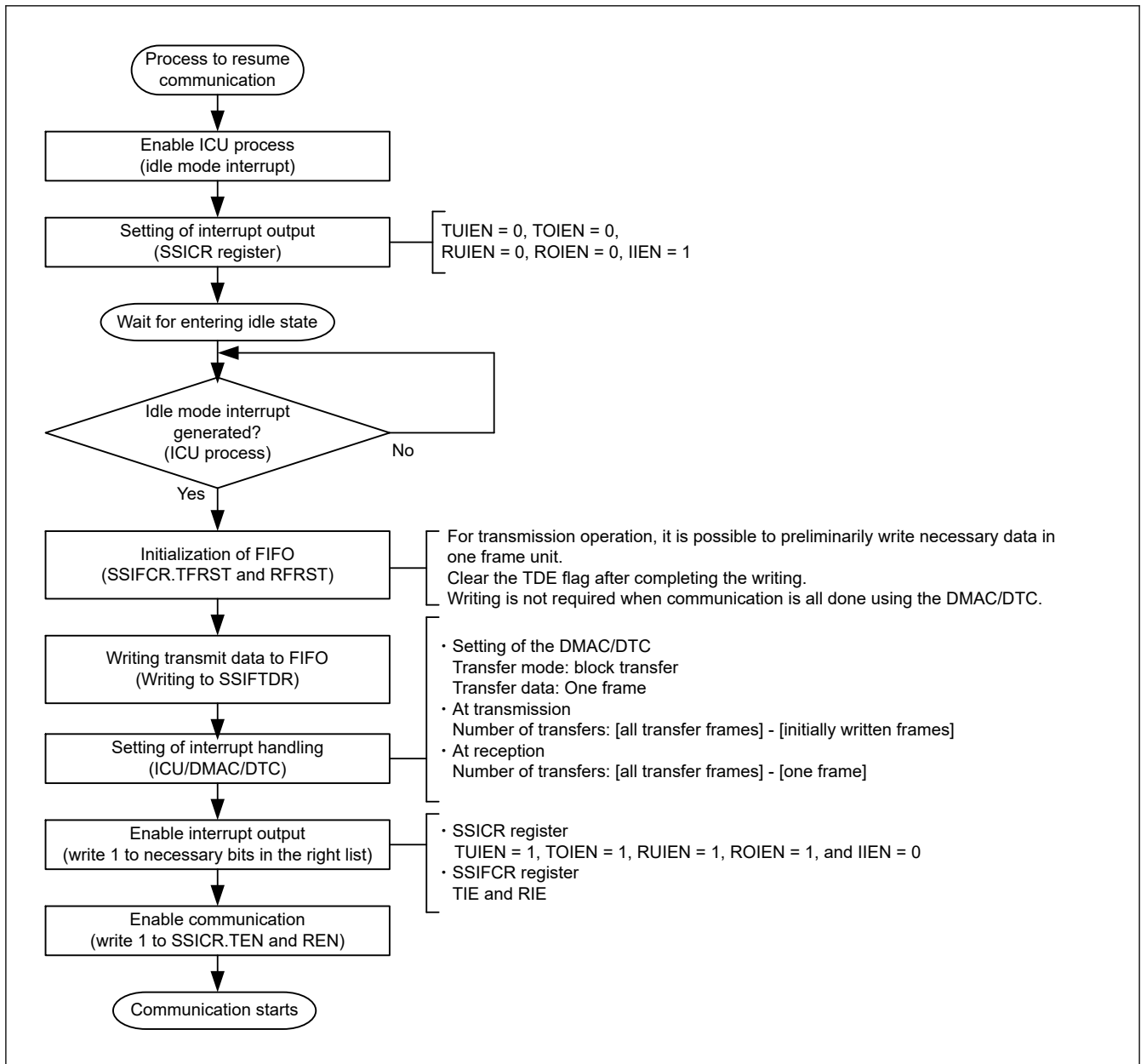
#### (4) Receive Overflow Error

If a receive overflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The receive data that caused the receive overflow error cannot be stored in the receive FIFO data register (SSIFRDR). To recover from the error, follow the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#).

### 37.6.7 Resume Communication

When you resume the communication using SSIE, follow the communication resume procedure in [Figure 37.58](#). The communication resume procedure is designed on the assumption that you resume the communication stopped by the

communication stop procedure without changing any settings. If you want to change clock and slave/master settings, use and follow the communication start procedure in [Figure 37.53](#). For details about the transmission operation and reception operation after starting communication, see [section 37.6.2. Transmission](#) and [section 37.6.3. Reception](#), respectively.



**Figure 37.58 Procedure to resume communication (CPU operation procedure)**

### 37.7 Interrupts

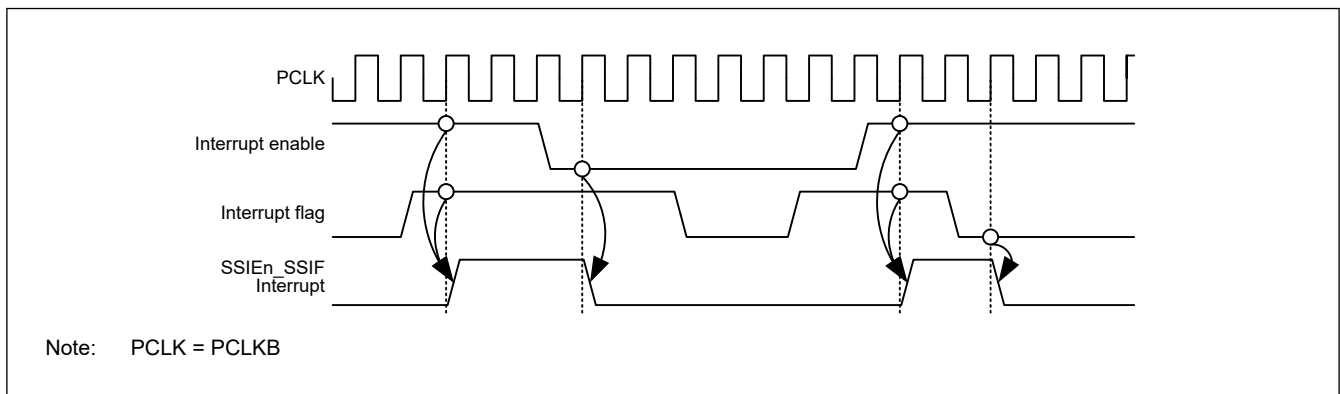
[Table 37.18](#) lists the interrupt sources. Set enable/disable of interrupt output of each source with the TUIEN, TOIEN, RUIEN, ROIEN, and I IEN bits in the SSICR register and the TIE and RIE bits in the SSIFCR register.

**Table 37.18 SSIE interrupt sources**

Channel	Interrupt source	Description	Interrupt flag	DMAC/DTC activation
SSIE0	SSIE0_SSIF	<ul style="list-style-type: none"> <li>• Transmit underflow interrupt</li> <li>• Transmit overflow interrupt</li> <li>• Receive underflow interrupt</li> <li>• Receive overflow interrupt</li> <li>• Idle interrupt</li> </ul>	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	Not possible
	SSIE0_SSIRXI	Receive data full interrupt	SSIFSR.RDF	Possible
	SSIE0_SSITXI	Transmit data empty interrupt	SSIFSR.TDE	Possible
SSIE1	SSIE1_SSIF	<ul style="list-style-type: none"> <li>• Transmit underflow interrupt</li> <li>• Transmit overflow interrupt</li> <li>• Receive underflow interrupt</li> <li>• Receive overflow interrupt</li> <li>• Idle interrupt</li> </ul>	SSISR.TUIRQ SSISR.TOIRQ SSISR.RUIRQ SSISR.ROIRQ SSISR.IIRQ	Not possible
	SSIE1_SSIRT	<ul style="list-style-type: none"> <li>• Receive data full interrupt</li> <li>• Transmit data empty interrupt</li> </ul>	SSIFSR.RDF SSIFSR.TDE	Possible

### 37.7.1 SSIE<sub>n</sub>\_SSIF Interrupt (n = 0, 1)

This interrupt source combines five interrupts. Enable output of necessary interrupts before using SSIE. The five interrupts are operated by using the flags assigned to individual interrupts and interrupt output enable bits. To clear an interrupt, set the interrupt enable to 0 or clear the interrupt flag to 0.



**Figure 37.59 Timing Diagram of the common interrupt source, SSIE<sub>n</sub>\_SSIF**

- Transmit underflow interrupt

As the transmit underflow interrupt, SSISR.TUIRQ is output while SSICR.TUIEN = 1. When you use SSIE for transmission, enable the output of this interrupt (SSICR.TUIEN = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#).

- Transmit overflow interrupt

As the transmit overflow interrupt, SSISR.TOIRQ is output while SSICR.TOIEN = 1. When you use SSIE for transmission, enable the output of this interrupt (SSICR.TOIEN = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#).

- Receive underflow interrupt

As the receive underflow interrupt, SSISR.RUIRQ is output while SSICR.RUIEN = 1. When you use SSIE for reception, enable the output of this interrupt (SSICR.RUIEN = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#).

- Receive overflow interrupt

As the receive overflow interrupt, SSISR.ROIRQ is output while SSICR.ROIEN = 1. When you use SSIE for reception, enable the output of this interrupt (SSICR.ROIEN = 1). If this interrupt occurs, follow instructions in the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#).

- Idle mode interrupt

As the idle mode interrupt, SSISR.IIRQ is output while SSICR.IIEN = 1. This interrupt is used to make sure that communication has stopped fully.

### 37.7.2 SSIE0\_SSITXI Interrupt [Full-duplex communication]

The transmit data empty interrupt is a pulse interrupt that is output when the following condition is met:

- SSIFCR.TIE = 1 and SSIFSR.TDE = 1  
 SSIE operation: When the value of SSIFSR.TDE changes from 0 to 1 while the value of SSIFCR.TIE is 1  
 CPU instruction: When the value of SSIFCR.TIE changes from 0 to 1 while the value of SSIFSR.TDE is 1

This interrupt is subject to the interrupt suppression function. If an interrupt condition for this interrupt occurs when the DTC/DMAC is busy (when the DTC/DMAC cannot accept interrupts), the interrupt suppression function holds the output of this interrupt. The held interrupt will be output after the DTC/DMAC is enabled to accept interrupts. For details, see [Figure 37.60](#).

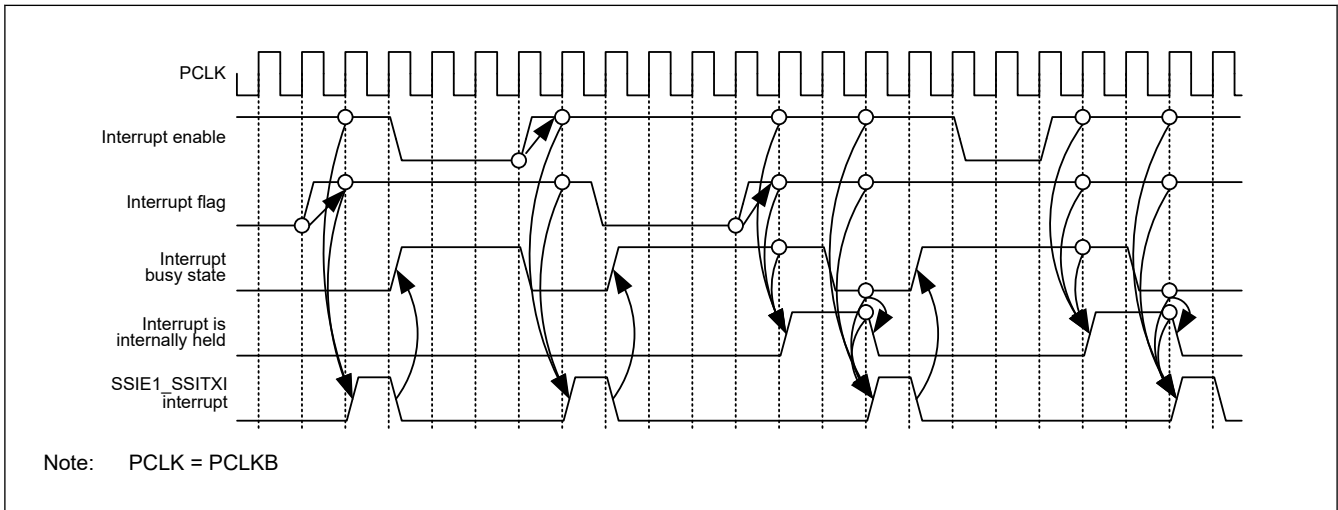


Figure 37.60 SSIE1\_SSITXI interrupt timing diagram

### 37.7.3 SSIE0\_SSIRXI Interrupt [Full-duplex communication]

The receive data full interrupt is a pulse interrupt that is output when the following condition is met:

- SSIFCR.RIE = 1 and SSIFSR.RDF = 1.  
 SSIE operation: When the value of SSIFSR.RDF changes from 0 to 1 while the value of SSIFCR.RIE is 1  
 CPU instruction: When the value of SSIFCR.RIE changes from 0 to 1 while the value of SSIFSR.RDE is 1

This interrupt is subject to the interrupt suppression function. If an interrupt condition for this interrupt occurs when the DTC/DMAC is busy (when the DTC/DMAC cannot accept interrupts), the interrupt suppression function holds the output of this interrupt. The held interrupt will be output after the DTC/DMAC is enabled to accept interrupts. The behavior of this interrupt is the same as the behavior shown in [Figure 37.60](#).

### 37.7.4 SSIE1\_SSIRT Interrupt [Half-duplex communication]

This interrupt is output by two sources, transmit data empty interrupt and receive data full interrupt. When this interrupt is generated, read the interrupt flag and specify the interrupt source.

This interrupt is subject to the interrupt suppression function. If an interrupt condition for this interrupt occurs when the DTC/DMAC is busy (when the DTC/DMAC cannot accept interrupts), the interrupt suppression function holds the output of this interrupt. The held interrupt will be output after the DTC/DMAC is enabled to accept interrupts. For details, see [Figure 37.61](#).



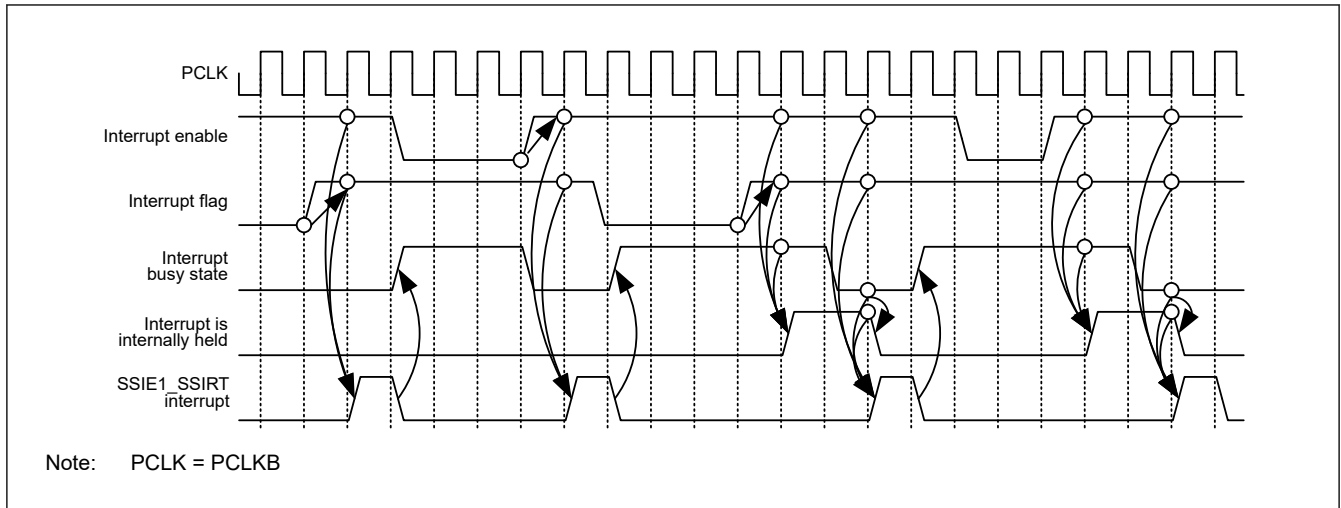


Figure 37.61 SSIE1\_SSIRT interrupt timing diagram

## 37.8 Software Resets

SSIE has three software reset bits to reset its states.

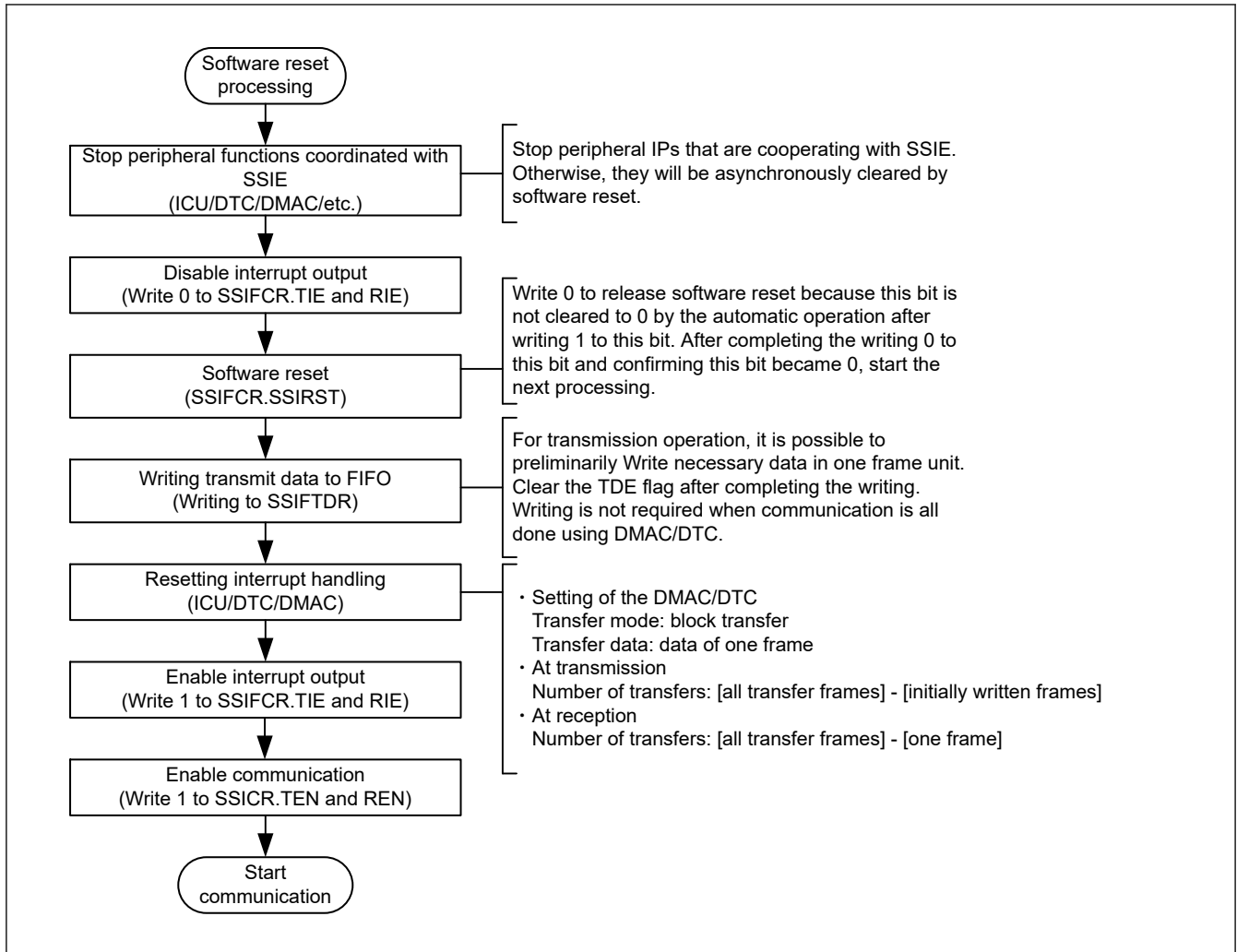
- SSIE software reset (SSIFCR.SSIRST)
- Transmit FIFO data register reset (SSIFCR.TFRST)
- Receive FIFO data register reset (SSIFCR.RFRST).

This section describes the procedures for the three types of software resets.

### 37.8.1 Software Reset Procedure

#### (1) SSIE Software Reset

For the SSIE software reset bit (SSIFCR.SSIRST), follow the procedure shown in [Figure 37.62](#). After a reset, the same setting is applied when it is resumed. To change the settings of clocks and slave/master mode, follow the procedure to start communication in [Figure 37.53](#). See [section 37.6.2. Transmission](#) and [section 37.6.3. Reception](#) respectively for transmission and reception after communication is resumed.



**Figure 37.62 Software reset procedure (CPU operation procedure)**

## (2) Transmit FIFO data register reset

To perform a transmit FIFO data register reset, follow instructions in the procedure to start communication in [Figure 37.53](#) and procedure to resume communication in [Figure 37.58](#).

## (3) Receive FIFO data register reset

To perform a receive FIFO data register reset, follow instructions in the procedure to start communication in [Figure 37.53](#) and procedure to resume communication in [Figure 37.58](#).

## 37.9 Usage Notes

### 37.9.1 Notes for Slave-mode Communication

#### 37.9.1.1 SSIBCKn (n = 0, 1) control

In slave-mode communication (SSICR.MST = 0), SSIE needs supply of SSIBCKn (n = 0, 1). To stop BCK on the master side, make sure that SSIE is in the idle state (SSISR.IIRQ = 1). If BCK is stopped before SSIE becomes idle, take the procedure to start communication in [Figure 37.53](#) or wait for an idle state by taking the procedure to resume communication in [Figure 37.58](#).

### 37.9.1.2 SSILRCKn/SSIFS<sub>n</sub> (n = 0, 1) pin

SSIE has the SSILRCK<sub>n</sub>/SSIFS<sub>n</sub> (n = 0, 1) pin, which indicates the synchronization of communication. When SSIE is in slave mode (SSICR.MST = 0), the communication format SSIE uses must match that of the other-party device to communicate. SSIE uses the signal input by the SSILRCK<sub>n</sub>/SSIFS<sub>n</sub> (n = 0, 1) pin only as a trigger to start communication.

## 37.9.2 Notes for Master-mode Communication

### 37.9.2.1 AUCKE control

In master-mode communication (SSICR.MST = 1), SSIE operates with the audio clock (AUDIO\_MCK). To stop SSIE completely, make sure that SSIE is in the idle state (SSISR.IIRQ = 1) and then write 0 to SSIFCR.AUCKE.

### 37.9.2.2 LRCONT control

To stop the output to the SSILRCK<sub>n</sub>/SSIFS<sub>n</sub> (n = 0, 1) pin with SSIOFR.LRCONT when SSIE is in the idle state in master-mode communication (SSICR.MST = 1), note the following: The output stops when the value of the SSIOFR.LRCONT bit is changed from 1 to 0. Make sure that the other-party device is not affected. For details, see [Figure 37.44](#).

### 37.9.2.3 BCKASTP control

To stop the output to the SSIBCK<sub>n</sub> (n = 0, 1) pin with SSIOFR.BCKASTP in master-mode communication (SSICR.MST = 1) and while SSIE is in the idle state, note the following: The output stops when the value of the SSIOFR.BCKASTP bit is changed from 0 to 1. So, make sure that the other-party device is not affected. For details, see [Figure 37.45](#).

The BCKASTP bit cannot be used when the other-party device (which is a slave) requires the clock output from the SSIBCK<sub>n</sub> (n = 0, 1) pin before and during communication.

## 37.9.3 Notes for Communication Flow

### 37.9.3.1 When an error interrupt is generated

SSIE has the following four errors.

- Transmit underflow error
- Transmit overflow error
- Receive underflow error
- Receive overflow error

When an underflow error or overflow error is generated, SSIE need to be restarted. Follow the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#).

#### (1) Transmit Underflow Error

If a transmit underflow error occurs, review the number of times of writing data to the transmit FIFO data register (SSIFTDR) in response to a transmit data empty interrupt. After a transmit underflow error occurs, SSIE outputs 0s as data. To normally output the serial data written to the transmit FIFO data register (SSIFTDR) to the SSITXD0/SSIDATA1 pin, follow the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#). After this error occurs, serial data is consumed as usual. If you resume communication, write the serial data from the beginning.

#### (2) Transmit Overflow Error

If a transmit overflow error occurs, review the number of times of writing data to the Transmit FIFO Data Register (SSIFTDR) in response to transmit data empty interrupts. The serial data written to the Transmit FIFO Data Register (SSIFTDR) that caused the transmit overflow error becomes invalid. This error can occur regardless of whether a transmission operation is being done. To recover from the error, follow the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#). When you resume communication, deal with the invalid serial data appropriately.

### (3) Receive Underflow Error

If a receive underflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The values read from the receive FIFO data register (SSIFRDR) that caused the receive underflow error are undefined. This error can occur regardless of whether a reception operation is being done. To recover from the error, follow the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#).

### (4) Receive Overflow Error

If a receive overflow error occurs, review the number of times of reading data from the receive FIFO data register (SSIFRDR) in response to receive data full interrupts. The receive data that caused the receive overflow error cannot be stored in the receive FIFO data register (SSIFRDR). To recover from the error, follow the procedure to halt communication in [Figure 37.56](#) and error-handling procedure in [Figure 37.57](#).

## 37.9.3.2 Transmit data empty interrupt

The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.TDE to write data to SSIFTDR. The number of times of writing data to SSIFTDR by detecting the value 1 of SSIFSR.TDE must be in accordance with the free space size of the transmit FIFO data register specified by SSISCR.TDES. After as much transmit data as the free space size is written to SSIFTDR, the SSIFSR.TDE flag must be cleared. Continuous transmission is enabled by repeating data writing. If the SSIFSR.TDE flag is not cleared, the flag is not cleared automatically.

## 37.9.3.3 Receive data full interrupt

The communication flow defined in SSIE uses the DTC/DMAC. If you do not use the DTC/DMAC, perform polling of the value 1 of SSIFSR.RDF to read data from SSIFRDR. The number of times of reading data from SSIFRDR by detecting the value 1 of SSIFSR.RDF must be in accordance with the receive data storage capacity of the receive FIFO data register specified by SSISCR.RDFS. After received data is read from SSIFRDR, the SSIFSR.RDF flag must be cleared. Continuous reception is enabled by repeating data reading. If the SSIFSR.RDF flag is not cleared, the flag is not cleared automatically.

## 37.9.3.4 Switching transfer modes

1. For state transition from transmission, reception, and transmission and reception, disable transmission and reception (SSICR.TEN = 0, SSICR.REN = 0).
2. Confirm it is in the idle state (SSISR.IIRQ = 1).
3. In the idle state, set the SSICR.TEN bit or the SSICR.REN bit again and resume transfer.

## 37.9.3.5 Resume communication after halting SSIE

When communication of SSIE is halted according to the procedure to halt communication in [Figure 37.56](#), resume communication according to the procedure to resume communication in [Figure 37.58](#).

## 37.9.4 Write Access Restriction

### 37.9.4.1 SSICR register

If the TEN bit or REN bit is rewritten, make sure that the SSISR.IIRQ bit is in the desired status. If the value of the TEN or REN bit is changed by rewriting, subsequent operation is unpredictable. For example, when transmission or reception is enabled, check that SSISR.IIRQ is 0; when transmission or reception is disabled, check that SSISR.IIRQ is 1.

#### (1) TEN Bit and REN Bit

These bits enable/disable transmission and reception. When 1 is written to one of these bits, the corresponding communication operation starts in synchronization with a start trigger by the SSILRCKn/SSIFSn (n = 0, 1) signal. For details, see [section 37.6.2. Transmission](#), [section 37.6.3. Reception](#), and [section 37.6.4. Transmission and Reception](#). When 0 is written to this bit, the current communication operation stops at the next frame boundary. To use SSIE for both transmission and reception, always write 1 to these bits together. When stopping the communication using SSIE, always disable both transmission and reception (write 0 to the TEN and REN bits).

### 37.9.4.2 SSISR register

#### (1) Clearing TUIRQ and TOIRQ

After communication is enabled (by changing the value of SSICR.TEN bit from 0 to 1), the transmission error flags (TOIRQ and TUIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the transmission error flags might be unable to be read.

#### (2) Clearing RUIRQ and ROIRQ

After communication is enabled (by changing the value of SSICR.REN bit from 0 to 1), the reception error flags (RUIRQ and ROIRQ in the SSISR register) are cleared. If, however, the SSISR register is read continuously, the cleared status of the reception error flags might be unable to be read.

### 37.9.4.3 Communication state

Writing to the bits with orange-shaded area in [Table 37.19](#) is prohibited. If written, the operation performed immediately after writing is not guaranteed.

**Table 37.19 Bits protected from writing during communication**

Symbol	Address (BASE+)		+0								+1							
			31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
			15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SSICR	0x00	+0	—	CKS	TUI EN	TOI EN	RUI EN	ROI EN	IIEN	—	FRM[1:0]	DWL[2:0]		SWL[2:0]				
		+2	—	MST	BCK P	LRC KP	SPD P	SDT A	PDT A	DEL	CKDV[3:0]			MU EN	—	TEN	REN	
SSISR	0x04	+0	—	—	TUI RQ	TOI RQ	RUI RQ	ROI RQ	IIRQ	—	—	—	—	—	—	—	—	
		+2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
SSIFCR	0x10	+0	AUC KE	—	—	—	—	—	—	—	—	—	—	—	—	—	SSI RST	
		+2	—	—	—	—	BS W	—	—	—	—	—	—	—	TIE	RIE	TFR ST	RFR ST
SSIFSR	0x14	+0	—	—	TDC[5:0]					—	—	—	—	—	—	—	TDE	
		+2	—	—	RDC[5:0]					—	—	—	—	—	—	—	RDF	
SSIFTDR	0x18	+0	SSIFTDR[31:16]															
		+2	SSIFTDR[15:0]															
SSIFRDR	0x1C	+0	SSIFRDR[31:16]															
		+2	SSIFRDR[15:0]															
SSIOFR	0x20	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
		+2	—	—	—	—	—	—	—	BCK AST P	LRC ONT	—	—	—	—	—	—	OMOD[1:0]
SSISCR	0x24	+0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	
		+2	—	—	—	TDES[4:0]					—	—	—	RDFS[4:0]				

### 37.9.5 Module-stop function

SSIE operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The SSIE module is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

## 38. Cyclic Redundancy Check (CRC)

### 38.1 Overview

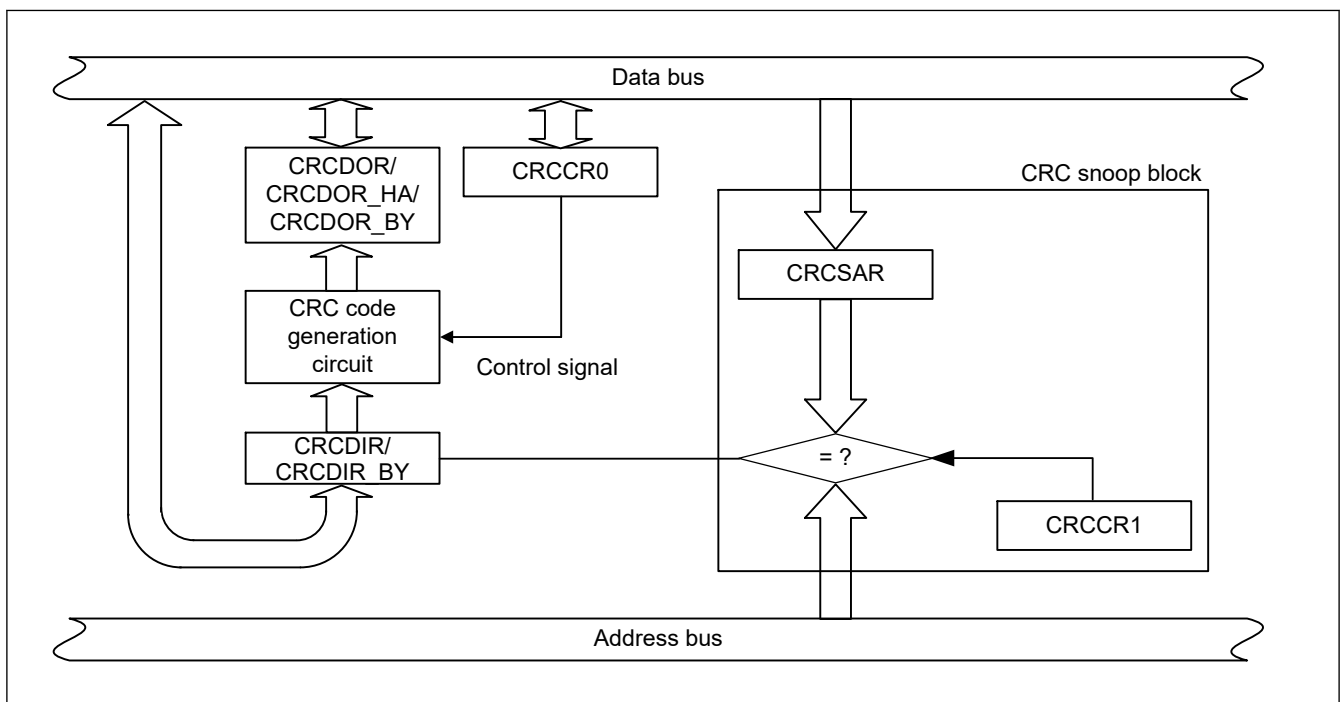
The Cyclic Redundancy Check (CRC) calculator generates CRC codes to detect errors in the data. The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication. Additionally, various CRC-generation polynomials are available. The snoop function allows monitoring reads from and writes to specific addresses. This function is useful in applications that require CRC code to be generated automatically in certain events, such as monitoring writes to the serial transmit buffer and reads from the serial receive buffer.

Table 38.1 lists the CRC calculator specifications and Figure 38.1 shows a block diagram.

**Table 38.1 CRC calculator specifications**

Item	Description	
Data size	8-bit	32-bit
Data for CRC calculation*1	CRC code generated for data in 8n-bit units (where n is a natural number)	CRC code generated for data in 32n-bit units (where n is a natural number)
CRC processor unit	Operation executed on 8 bits in parallel	Operation executed on 32 bits in parallel
CRC generating polynomial	One of three generating polynomials that is selectable: [8-bit CRC] <ul style="list-style-type: none"> <li><math>X^8 + X^2 + X + 1</math> (CRC-8)</li> </ul> [16-bit CRC] <ul style="list-style-type: none"> <li><math>X^{16} + X^{15} + X^2 + 1</math> (CRC-16)</li> <li><math>X^{16} + X^{12} + X^5 + 1</math> (CRC-CCITT)</li> </ul>	One of two generating polynomials that is selectable: [32-bit CRC] <ul style="list-style-type: none"> <li><math>X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1</math> (CRC-32)</li> <li><math>X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1</math> (CRC-32C)</li> </ul>
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB-first or MSB-first communication	
Module-stop function	Module-stop state can be set to reduce power consumption	
CRC snoop	Monitor reads from and writes to a certain register address	
TrustZone Filter	Security and Privilege attribution can be set	

Note 1. This function cannot divide data used in CRC calculations. Write data in 8-bit or 32-bit units.



**Figure 38.1 CRC calculator block diagram**

## 38.2 Register Descriptions

### 38.2.1 CRCCR0 : CRC Control Register 0

Base address: CRC = 0x4031\_0000  
CRC\_NS = 0x5031\_0000

Offset address: 0x00

Bit position: 7 6 5 4 3 2 1 0

Bit field:	DORCLR	LMS	—	—	—	GPS[2:0]	
------------	--------	-----	---	---	---	----------	--

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	GPS[2:0]	CRC Generating Polynomial Switching 0 0 1: 8-bit CRC-8 ( $X^8 + X^2 + X + 1$ ) 0 1 0: 16-bit CRC-16 ( $X^{16} + X^{15} + X^2 + 1$ ) 0 1 1: 16-bit CRC-CCITT ( $X^{16} + X^{12} + X^5 + 1$ ) 1 0 0: 32-bit CRC-32 ( $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ ) 1 0 1: 32-bit CRC-32C ( $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ ) Others: No calculation is executed	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	LMS	CRC Calculation Switching 0: Generate CRC code for LSB-first communication 1: Generate CRC code for MSB-first communication	R/W
7	DORCLR	CRCDOR/CRCDOR_HA/CRCDOR_BY Register Clear 0: No effect 1: Clear the CRCDOR/CRCDOR_HA/CRCDOR_BY register	W

Note: S-TYPE3, P-TYPE3

#### GPS[2:0] bits (CRC Generating Polynomial Switching)

The GPS[2:0] bits select the CRC generating polynomial.

#### LMS bit (CRC Calculation Switching)

The LMS bit selects the bit order of generated CRC code. Transmit the lower byte of the CRC code first for LSB-first communication and the upper byte first for MSB-first communication. For details on transmitting and receiving CRC code, see [section 38.3. Operation](#).

#### DORCLR bit (CRCDOR/CRCDOR\_HA/CRCDOR\_BY Register Clear)

Write 1 to the DORCLR bit to set the CRCDOR/CRCDOR\_HA/CRCDOR\_BY register to 0x00000000. This bit is read as 0. Only 1 can be written to it.

### 38.2.2 CRCCR1 : CRC Control Register 1

Base address: CRC = 0x4031\_0000  
CRC\_NS = 0x5031\_0000

Offset address: 0x01

Bit position: 7 6 5 4 3 2 1 0

Bit field:	CRCS EN	CRCS WR	—	—	—	—	—
------------	---------	---------	---	---	---	---	---

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	CRCSWR	Snoop-On-Write/Read Switch 0: Snoop-on-read 1: Snoop-on-write	R/W
7	CRCSEN	Snoop Enable 0: Disabled 1: Enabled	R/W

Note: S-TYPE3, P-TYPE3

### CRCSWR bit (Snoop-On-Write/Read Switch)

The CRCSWR bit selects the direction of access in the CRC snoop function.

When this bit is set to 0 (initial value), the CRC snoop operation to read a specific register is enabled. Similarly, when this bit is set to 1, the CRC snoop operation to write a specific register is enabled.

### CRCSEN bit (Snoop Enable)

When the CRCSEN bit is set to 1, the CRC snoop operation is enabled. When this bit is set to 0, the CRC snoop operation is disabled.

### 38.2.3 CRCDIR/CRCDIR\_BY : CRC Data Input Register

Base address: CRC = 0x4031\_0000  
CRC\_NS = 0x5031\_0000

Offset address: 0x04

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	CRC input data The CRCDIR register is a 32-bit read/write register to write data for CRC-32 or CRC-32C calculation. The CRCDIR_BY (CRCDIR[31:24]) is an 8-bit read/write register to write data for CRC-8, CRC-16, or CRC-CCITT calculation.	R/W

Note: S-TYPE3, P-TYPE3

### 38.2.4 CRCDOR/CRCDOR\_HA/CRCDOR\_BY : CRC Data Output Register

Base address: CRC = 0x4031\_0000  
CRC\_NS = 0x5031\_0000

Offset address: 0x08

Bit position: 31 0



Value after reset: 0



Bit	Symbol	Function	R/W
31:0	n/a	CRC output data The CRCDOR register is a 32-bit read/write register for CRC-32 or CRC-32C calculation. The CRCDOR_HA (CRCDOR[31:16]) register is a 16-bit read/write register for CRC-16 or CRC-CCITT calculation. The CRCDOR_BY (CRCDOR[31:24]) register is an 8-bit read/write register for CRC-8 calculation. Because its initial value is 0x00000000, rewrite the CRCDOR/CRCDOR_HA/CRCDOR_BY register to perform the calculations using a value other than the initial value. Data written to the CRCDIR/CRCDIR_BY register is CRC calculated and the result is stored in the CRCDOR/CRCDOR_HA/CRCDOR_BY register. If the CRC code is calculated following the transferred data and the result is 0x00000000, there is no CRC error.	R/W

Note: S-TYPE3, P-TYPE3

### 38.2.5 CRCSAR : Snoop Address Register

Base address: CRC = 0x4031\_0000  
CRC\_NS = 0x5031\_0000

Offset address: 0x0C



Bit	Symbol	Function	R/W
13:0	CRCSA[13:0]	Register Snoop Address These bits store the TDR or RDR address in the SCI module to snoop	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

#### CRCSA[13:0] bits (Register Snoop Address)

The CRCSA[13:0] bits specify the lower address 14 bits of the register monitored by the CRC snoop operation.

Only the following addresses can be used for the CRCSA[13:0] bits:

[secure region address]

- 0x4035\_8004: SCI0.TDR, 0x4035\_8000: SCI0.RDR
- 0x4035\_8104: SCI1.TDR, 0x4035\_8100: SCI1.RDR
- 0x4035\_8204: SCI2.TDR, 0x4035\_8200: SCI2.RDR
- 0x4035\_8304: SCI3.TDR, 0x4035\_8300: SCI3.RDR
- 0x4035\_8404: SCI4.TDR, 0x4035\_8400: SCI4.RDR
- 0x4035\_8904: SCI9.TDR, 0x4035\_8900: SCI9.RDR

[non-secure region address]

- 0x5035\_8004: SCI0.TDR, 0x5035\_8000: SCI0.RDR
- 0x5035\_8104: SCI1.TDR, 0x5035\_8100: SCI1.RDR
- 0x5035\_8204: SCI2.TDR, 0x5035\_8200: SCI2.RDR
- 0x5035\_8304: SCI3.TDR, 0x5035\_8300: SCI3.RDR
- 0x5035\_8404: SCI4.TDR, 0x5035\_8400: SCI4.RDR
- 0x5035\_8904: SCI9.TDR, 0x5035\_8900: SCI9.RDR

### 38.3 Operation

#### 38.3.1 Basic Operation

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer.

The following examples show CRC code generation for input data (0xF0) using the 16-bit CRC-CCITT generating polynomial ( $X^{16} + X^{12} + X^5 + 1$ ). In these examples, the value of the CRC Data Output Register (CRCDOR\_HA) is cleared before CRC calculation.

When an 8-bit CRC (with the polynomial  $X^8 + X^2 + X + 1$ ) is in use, the valid bits of the CRC code are obtained in CRCDOR\_BY. When a 32-bit CRC is in use, the valid bits of the CRC code are obtained in CRCDOR.

Figure 38.2 and Figure 38.3 show the LSB-first and MSB-first data transmission examples respectively. Figure 38.4 and Figure 38.5 show the LSB-first and MSB-first data reception examples.

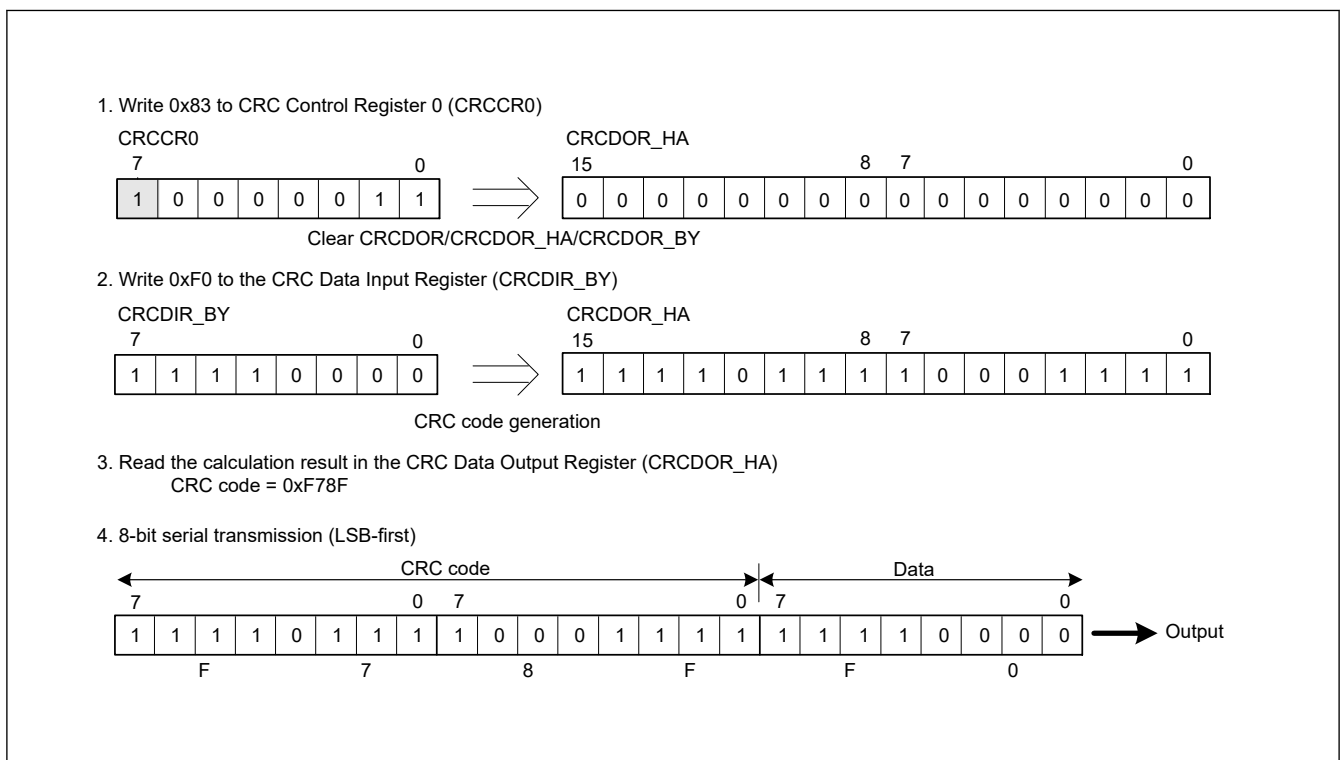


Figure 38.2 LSB-first data transmission

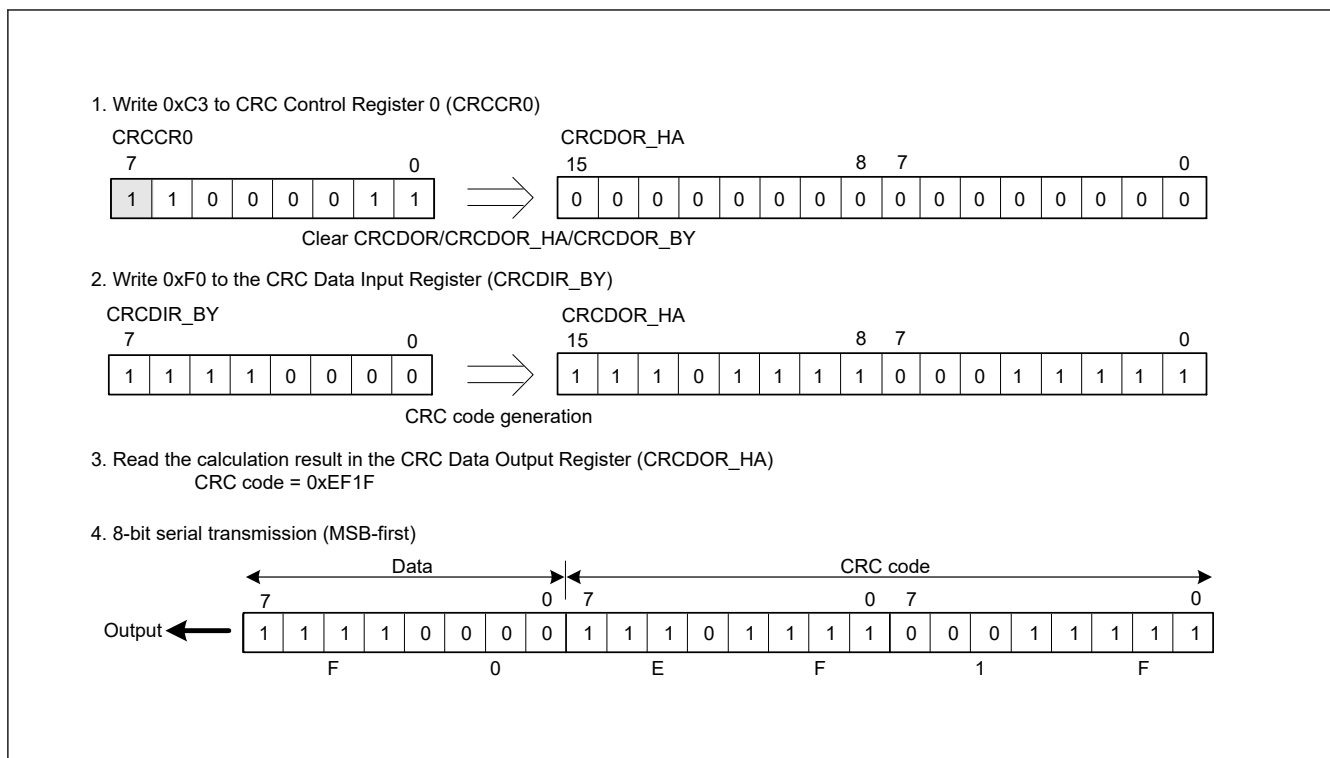


Figure 38.3 MSB-first data transmission

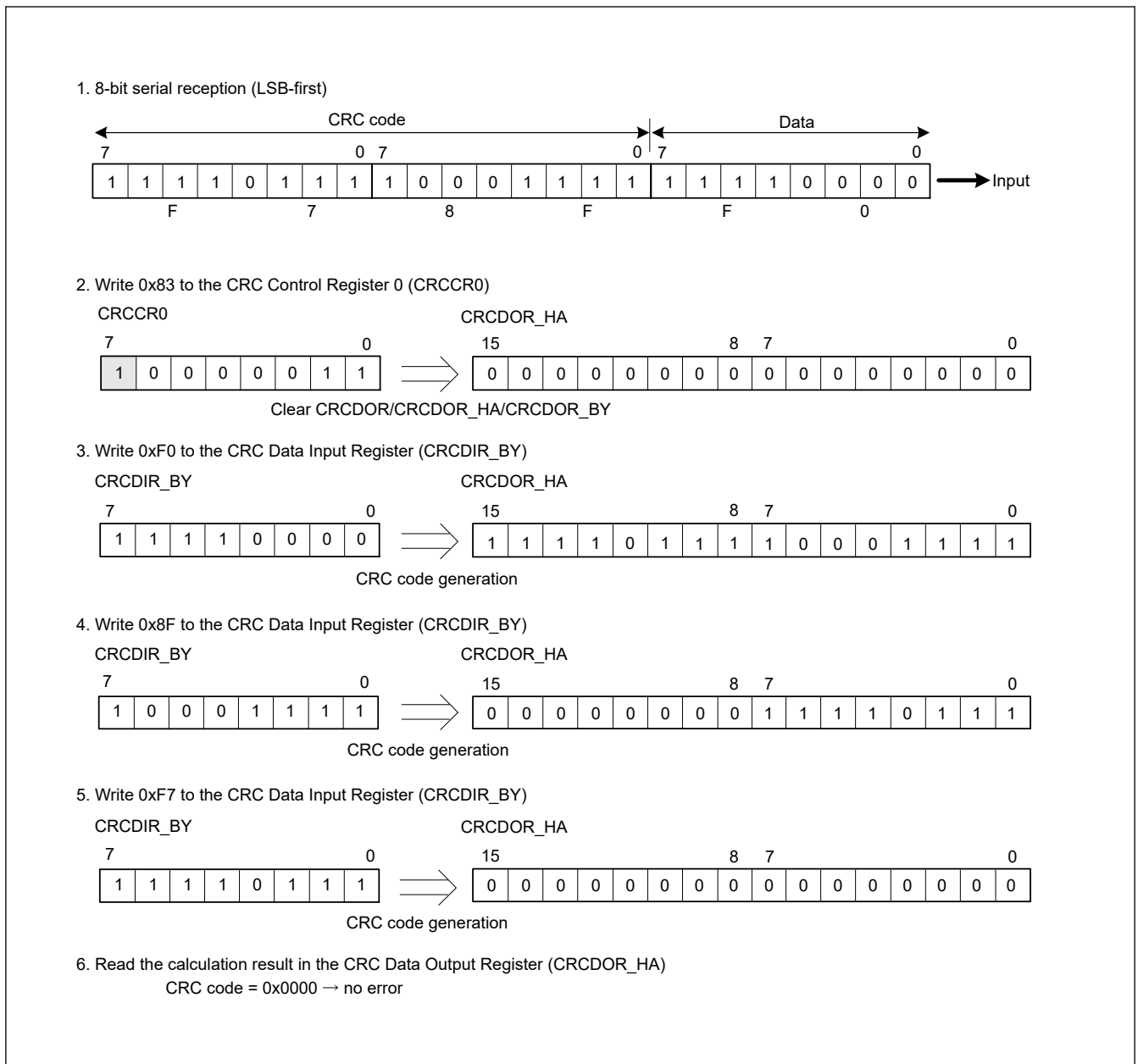


Figure 38.4 LSB-first data reception

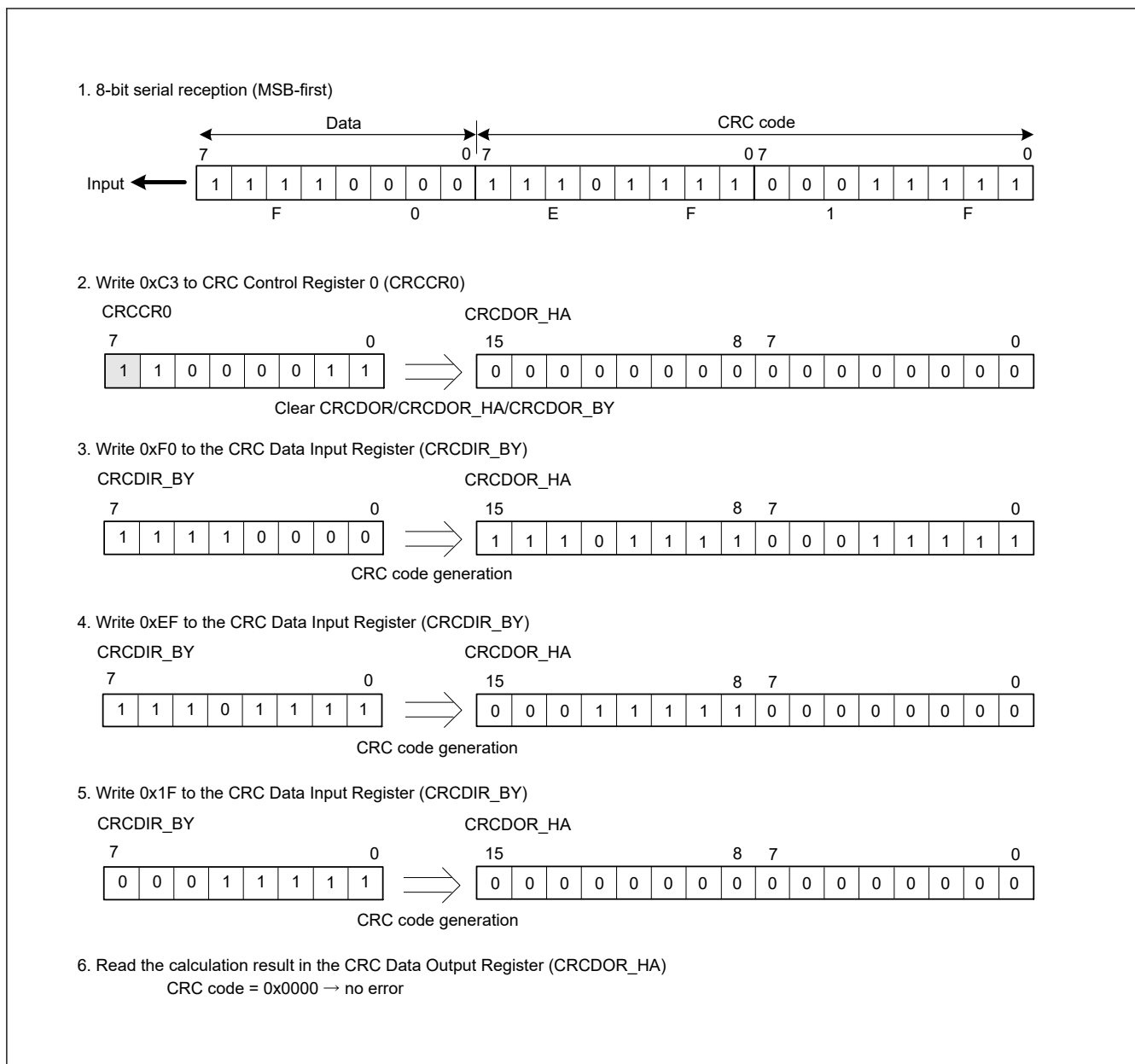


Figure 38.5 MSB-first data reception

### 38.3.2 CRC Snoop Function

The CRC snoop function monitors reads from and writes to a specific register and performs CRC calculation on the monitored data automatically. Because the CRC snoop function recognizes writes to and reads from a specific register address as a trigger to automatically perform CRC calculation, there is no need to write data to the CRCDIR register. All I/O register specified in the [section 38.2.5. CRCSAR : Snoop Address Register](#) are subject to the CRC snoop. The CRC snoop is useful in monitoring writes to the SCIn.TDR (n = 0 to 4, 9) register, and reads from the SCIn.RDR (n = 0 to 4, 9) register.

To use this function, write the lower address 14 bits of a specific register to bits CRCSA13 to CRCSA0 in the CRCSAR register, and set CRCSEN bit in the CRCCR1 register to 1. Then, set the CRCSWR bit in the CRCCR1 register to 1 to enable snooping on writes to the target register, or set the CRCSWR bit in the CRCCR1 register to 0 to enable snooping on reads from the target register. It is possible that access to a target I/O register may be executed before the CRCSWR bit write completed. In this case, the data is not stored in the CRCDIR register. To avoid this issue, before accessing I/O register, read back the CRCSWR bit that was written to confirm that the write completed.

When both the CRCSEN and CRCSWR bits are set to 1, and data is written to a target register in a bus master module such as the CPU, DMAC, and DTC, the CRC calculator stores the data in the CRCDIR register and performs CRC calculation. Similarly, when the CRCSEN bit is set to 1, CRCSWR bit to 0, and data is read from a target register in a bus master

module such as the CPU, DMAC, and DTC, the CRC calculator stores the data in the CRCDIR register and performs CRC calculations.

When the CRC code is generated by using CRC-8, CRC-16, and CRC-CCITT generating polynomial, the target register is accessed in 1 byte (8 bits). RDR\_BY and TDR\_LL should be used to access RDR and TDR. Similarly, when the CRC code is generated by using CRC-32 and CRC-32C generating polynomial, the target register is accessed in words (32 bits). Note that for RDR and TDR, CRC codes are generated that contain data other than RDAT and TDAT.

When CRC is marked as secure by PSARC.PSARC1 bit, the CRC snoop function is available for secure access to the specified I/O registers. When CRC is marked as non-secure by PSARC.PSARC1 bit, the CRC snoop function is available for non-secure access to the specified I/O registers.

## 38.4 Usage Notes

### 38.4.1 Settings for the Module-Stop State

The Module Stop Control Register C (MSTPCRC) can enable or disable CRC calculator operation. The CRC calculator is initially stopped after a reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

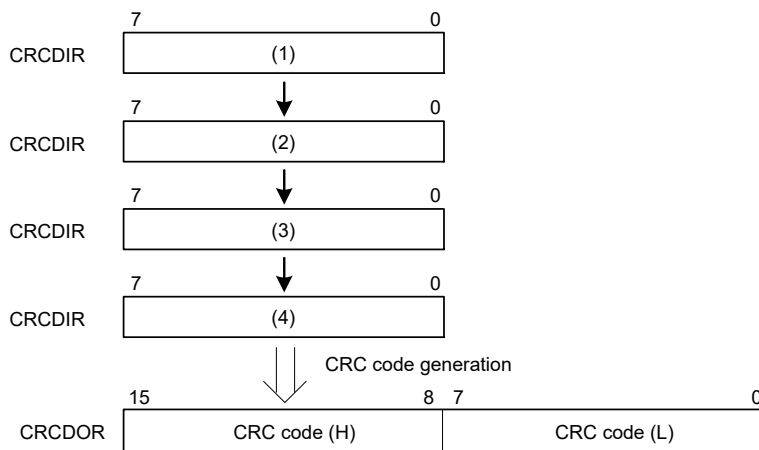
### 38.4.2 Note on Transmission

The transmission sequence for the CRC code differs based on whether the transmission is LSB-first or MSB-first. [Figure 38.6](#) shows an LSB-first and MSB-first data transmission.

When transmitting 32-bit data (for operation executed on 8 bits in parallel)

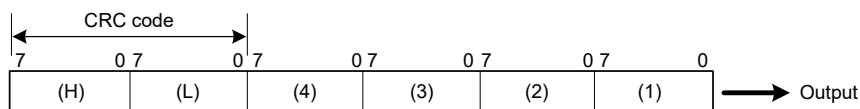
1. CRC code

After specifying the method for generation calculation, write data to CRCDIR in order of (1), (2), (3), and (4).



2. Transmit data

(i) When transmission is LSB-first



(ii) When transmission is MSB-first

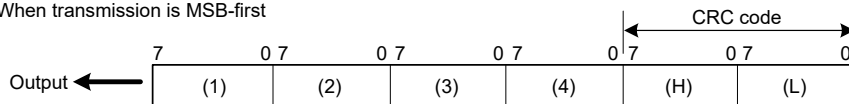


Figure 38.6 LSB-first and MSB-first data transmission

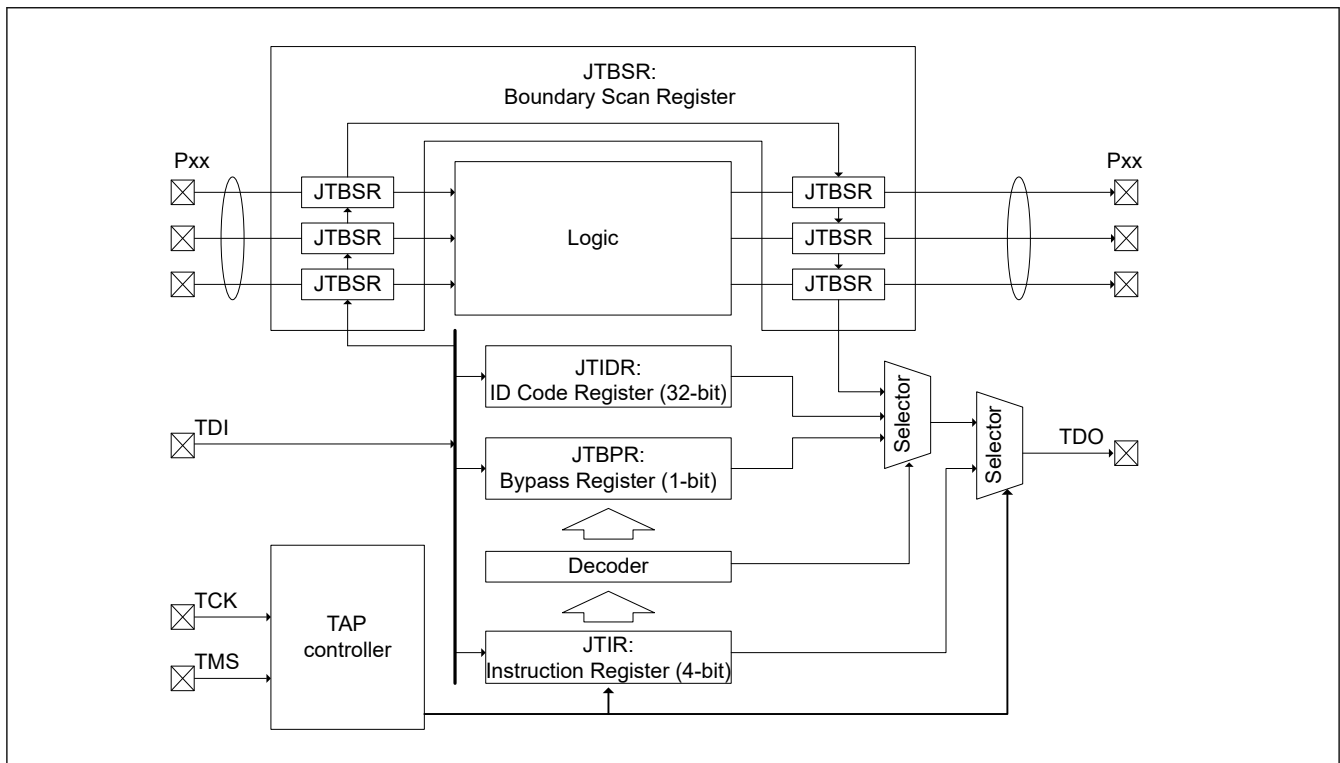
## 39. Boundary Scan

### 39.1 Overview

The boundary scan function provides a serial I/O interface based on the JTAG (Joint Test Action Group), IEEE Std.1149.1, and IEEE Standard Test Access Port and Boundary Scan Architecture. [Table 39.1](#) lists the boundary scan specifications, [Figure 39.1](#) shows a block diagram, and [Table 39.2](#) lists the I/O pins.

**Table 39.1** Boundary scan specifications

Parameter	Specifications
Execution condition	Boundary scan must be executed when the RES pin is driven low.
Test modes	<ul style="list-style-type: none"> <li>• BYPASS mode</li> <li>• EXTEST mode</li> <li>• SAMPLE/PRELOAD mode</li> <li>• CLAMP mode</li> <li>• HIGHZ mode</li> <li>• IDCODE mode</li> </ul>



**Figure 39.1** Boundary scan function block diagram

**Table 39.2** Boundary scan I/O pins

Pin name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. The input clock duty cycle is 50% when the boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin

Note: This device does not support the TRST pin for the JTAG interface.

### 39.2 Register Descriptions

[Table 39.3](#) lists the boundary scan registers.



**Table 39.3** Boundary scan registers

Register name	Symbol	Value after reset
Instruction Register	JTIR	0xE
ID Code Register	JTIDR	0x0841_9447
Bypass Register	JTBPR	Undefined
Boundary Scan Register	JTBSR	Undefined

Usage notes for the boundary scan registers:

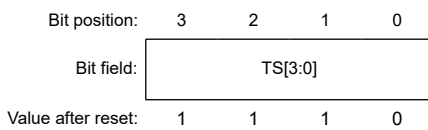
- Instructions can be input to the Instruction Register (JTIR) through the TDI pin by serial transfer.
- The Bypass Register (JTBPR), which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.
- The Boundary Scan Register (JTBSR), which is configured according to the BSDL description, is connected between the TDI and TDO pins when test data is being shifted in.

Table 39.4 shows the availability of serial transfer for the registers.

**Table 39.4** Serial transfer for registers

Register name	Serial input	Serial output
Instruction Register (JTIR)	Available	Available
ID Code Register (JTIDR)	Available	Available
Bypass Register (JTBPR)	Available	Available
Boundary Scan Register (JTBSR)	Available	Available

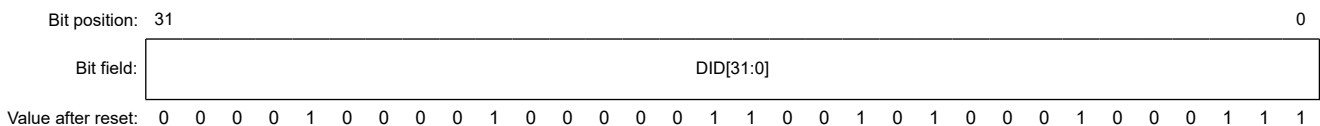
### 39.2.1 JTIR : Instruction Register



Bit	Symbol	Function	R/W																
3:0	TS[3:0]	Test Bit Set The command configuration for these bits	—																
		<table border="1" style="width: 100%;"><thead><tr> <th>TS[3:0]</th> <th>Instruction</th> </tr></thead><tbody> <tr><td>0x0</td><td>EXTEST</td></tr> <tr><td>0x1</td><td>SAMPLE/PRELOAD</td></tr> <tr><td>0x3</td><td>IDCODE (Renesas code)</td></tr> <tr><td>0x5</td><td>CLAMP</td></tr> <tr><td>0x6</td><td>HIGHZ</td></tr> <tr><td>0xF</td><td>BYPASS</td></tr> <tr><td>Others</td><td>Reserved</td></tr> </tbody></table>		TS[3:0]	Instruction	0x0	EXTEST	0x1	SAMPLE/PRELOAD	0x3	IDCODE (Renesas code)	0x5	CLAMP	0x6	HIGHZ	0xF	BYPASS	Others	Reserved
		TS[3:0]		Instruction															
		0x0		EXTEST															
		0x1		SAMPLE/PRELOAD															
		0x3		IDCODE (Renesas code)															
		0x5		CLAMP															
		0x6		HIGHZ															
		0xF		BYPASS															
Others	Reserved																		

JTAG instructions can be transferred to the JTIR register by serial input from the TDI pin. The JTIR register is initialized when a power-on reset occurs, or when the TAP controller is in the Test-Logic-Reset state.

### 39.2.2 JTIDR : ID Code Register



Bit	Symbol	Function	R/W
31:0	DID[31:0]	Device ID These bits store the fixed value that indicates the device IDCODE (0x0841_9447).	—

The JTIDR register data is output from the TDO pin when the IDCODE instruction is executed. After a reset release, the DID[31:0] of JTIDR changes into the Arm® debug code. See the *Arm® CoreSight™ SoC-400 Technical Reference Manual* (ARM DDI 0480F).

### 39.2.3 JTBPR : Bypass Register

The JTBPR register is a 1-bit register and is connected between the TDI and TDO pins when the JTIR register is set to BYPASS mode. The JTBPR register cannot be read from or written to by the CPU.

### 39.2.4 JTBSR : Boundary Scan Register

The JTBSR register is a shift register for controlling the external input and output pins of this device, and is distributed across the pads. To apply the JTBSR register in boundary-scan testing, issue the EXTEST, SAMPLE/PRELOAD, CLAMP and HIGHZ instructions. The BSDL file describes the associations between the JTBSR register bits and the pins of this device. The value after reset is undefined.

## 39.3 Operation

During a reset, the JTAG ports, TCK, TMS, TDI and TDO are assigned as default pin functions. The TCK, TMS and TDI pins are pulled up by the pull-up resistors. Boundary scan testing can be executed after the setup time elapses when POR is negated and RES is driven low.

### 39.3.1 TAP Controller

[Figure 39.2](#) shows the state transition diagram of the TAP controller. All transitions are controlled by the TMS signal.

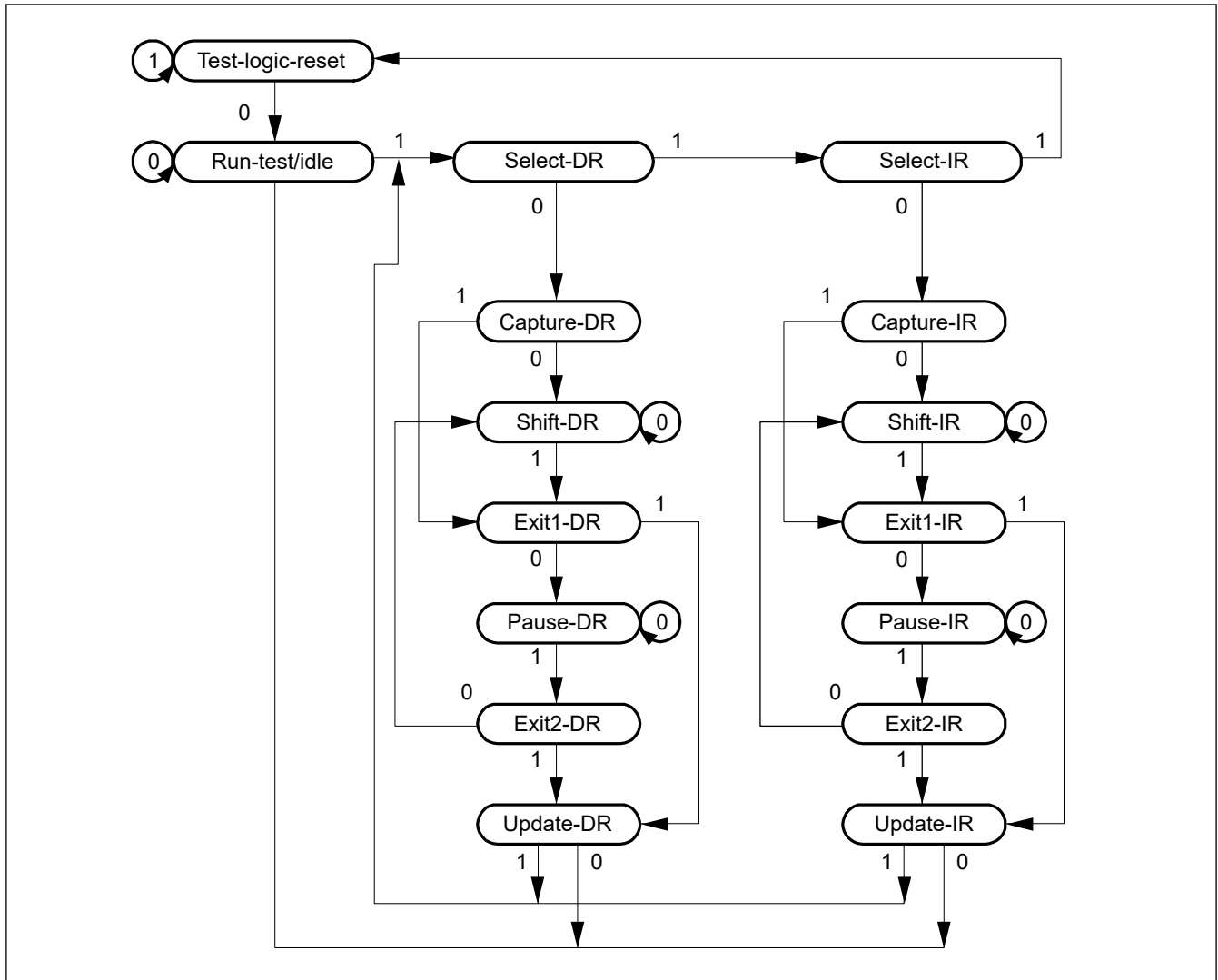


Figure 39.2 State transition diagram of TAP controller

### 39.3.2 Commands

#### (1) BYPASS

The BYPASS instruction drives the Bypass Register (JTBPB). This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed circuit board at higher speeds. While this instruction is being executed, the test circuit has no effect on the system circuits.

The JTBPB register is connected between the TDI and TDO pins. Bypass operation is initiated from the Shift-DR operation. The TDO is low in the first clock cycle in the Shift-DR state. In the subsequent clock cycles, values input to the TDI pin are output from the TDO pin.

#### (2) EXTEST

The EXTEST instruction is used to test external circuits when this device is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified in the SAMPLE/PRELOAD instruction) from the Boundary Scan Register (JTBSR) to the other devices, and input pins are used to input the test result.

#### (3) SAMPLE/PRELOAD

The SAMPLE/PRELOAD instruction is used to input data from the internal circuits of this device to the JTBSR register, output data from the scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to this device and output signals are also directly output to the external circuits. This device system circuit is not affected by this instruction.

In SAMPLE operation, the JTBSR register latches a snapshot of the data transferred from the input pins to the internal circuit or data transferred from the internal circuit to the output pins. The latched data is read from the scan path. The JTBSR register latches the data snapshot on the rising edge of the TCK pin in the Capture-DR state. The data snapshot is only transferred from the internal circuit to the output pins during a reset.

In PRELOAD operation, the initial value is written from the scan path to the parallel output latch of the JTBSR register prior to the EXTEST instruction execution. If EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In the EXTEST instruction, output parallel latches are always output to the output pins.)

#### (4) IDCODE

When the IDCODE instruction is selected, the ID Code Register (JTIDR) value is output to the TDO pin in the Shift-DR state of the TAP controller. In this case, the JTIDR register value is output LSB-first. During this instruction execution, the test circuit does not affect the system circuit.

#### (5) CLAMP

When the CLAMP instruction is selected, output pins output the JTBSR register value that was specified in the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of the JTBSR register is maintained regardless of the TAP controller state.

The JTBPR register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

#### (6) HIGHZ

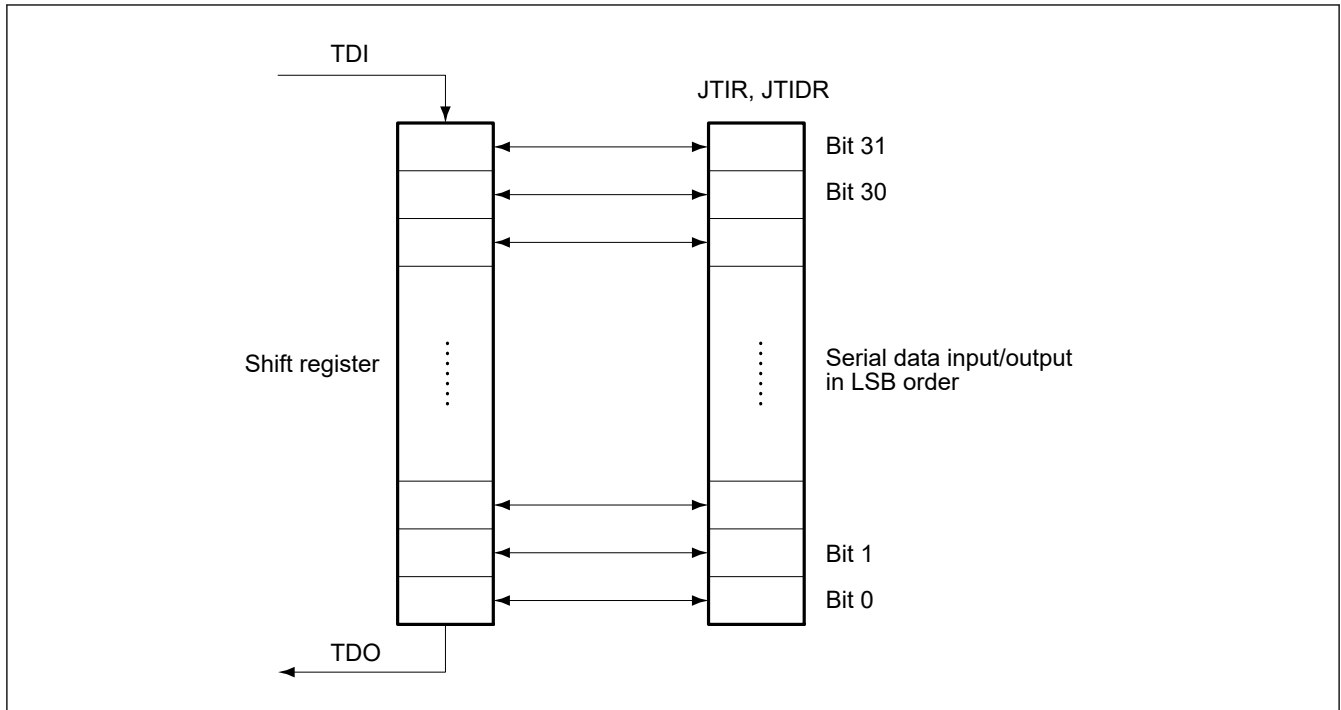
When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the JTBSR register is maintained regardless of the state of the TAP controller.

The JTBPR register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction is selected.

### 39.4 Usage Notes

The boundary scan function is subject to the following constraints:

- The boundary scan must be executed when the RES pin is driven low
- Serial data input/output is in LSB order, as shown in [Figure 39.3](#)



**Figure 39.3 Serial data input/output**

The following pins cannot be boundary-scanned:

- Power supply pins (VCC, VCC2, VCC\_DCDC, VCL, VSS, VSS\_DCDC, VBATT, AVCC0, AVSS0, VCC\_USB, VSS\_USB)
- Analog reference pins (VREFH0, VREFL0, VREFH, VREFL)
- Clock pins (EXTAL, XTAL, XCIN, and XCOU)
- Reset pin (RES)
- USBFS pin (USB\_DP, USB\_DM)
- The boundary-scan pins (TCK, TMS, TDI, and TDO).
- The switching regulator pin (VLO)

## 40. Security Features

### 40.1 Features

- Armv8-M TrustZone Technology
  - Implementation Defined Attribution Unit (IDAU) is implemented
  - Security Attribution Unit (SAU) is implemented
    - 8 regions
  - Master Security Attribution Unit (MSAU) is implemented (IDAU for master other than CPU)
  - Code flash
    - Up to 2 regions (secure/non-secure) in linear mode
    - Up to 4 regions (2 secure regions/2 non-secure regions) in dual mode
  - Data flash
    - Up to 2 regions (secure/non-secure)
  - SRAM
    - Up to 2 regions (secure/non-secure)
  - Standby SRAM
    - Up to 2 regions (secure/non-secure)
  - VBATT backup registers
    - Up to 2 regions (secure/non-secure)
  - Peripheral
    - Security attributes can be set individually for each unit/channel
  - OSPI\*1
    - OSPI region is defined as non-secure
- Privileged control
  - Access permissions for memory except Standby SRAM and VBATT backup register are controlled by MPU managed by the privileged code
  - Privilege attributes of Standby SRAM and VBATT backup register are controlled by the registers of each controller
  - Individual privileged or Unprivileged attribution for each peripheral.
- Device lifecycle management
- Three debug levels
  - AL2: Non-secure and secure debug functions are enabled and accessible from the debugger
  - AL1: Only non-secure debug functions is enabled, and the debugger can access only defined non-secure debug accessible regions
  - AL0: No debug functions are available.
- Key injection
- Secure factory programming
  - Supports image programming in ciphertext format in an untrusted factory
- Cryptographic accelerator
  - See [section 41, Renesas Secure IP \(RSIP-E51A\)](#).
- Secure pin multiplexing
  - All I/O port pins can be configured individually as secure or non-secure
  - Peripheral pin function is valid when the security attribute of peripheral and I/O port match. See [section 19, I/O Ports](#).

Note 1. There is no TrustZone filter in the external RAM and external device area. Therefore, even if these regions are marked as secure in the SAU settings, non-secure masters except the CPU can access them.

## 40.2 Tamper Detection

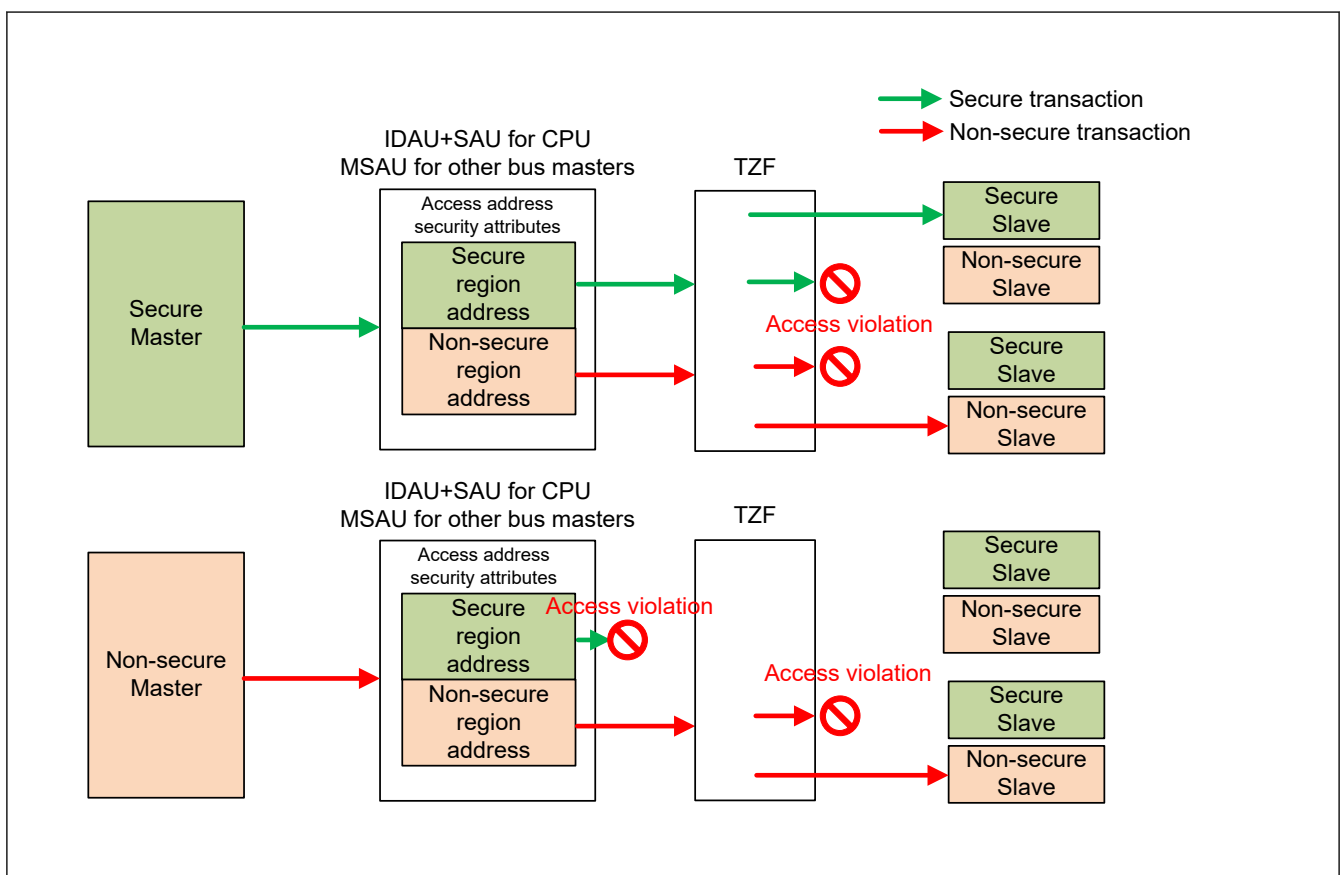
The device is provided with I/O ports capable of detecting an external tamper attempt. In the case of a tamper event:

- **Timestamping**  
The RTC can take a timestamp to log the event, and an interrupt can be generated. See [section 24, Realtime Clock \(RTC\)](#).
- **VBATT backup registers**  
VBATT backup registers are zeroized (set to zero) following detection of a tamper event. See [section 11, Battery Backup Function](#).

## 40.3 Arm Security Features

### 40.3.1 Arm TrustZone Technology

Arm TrustZone technology divides the system and the application into secure and non-secure domains. A secure application can issue both secure and non-secure transactions, but a non-secure application can only issue non-secure transactions. Secure transactions can only access secure memory and resources, and non-secure transactions can only access non-secure memory and resources. Secure transactions can be issued only using secure region addresses and non-secure transactions can be issued only using non-secure region addresses. [Figure 40.1](#) shows the security attributes of transactions that can be issued by each master.



**Figure 40.1 Transactions that can be issued by each master**

For more details, see the *Arm® v8-M Architecture Reference Manual and Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v8-M 1.0*.

### 40.3.2 Privileged Control

Systems and applications are divided into privileged and unprivileged domains. The CPU can limit or exclude access to some resources by executing code in privileged or unprivileged mode. Privileged mode can access both privileged and unprivileged domains, but unprivileged mode can access only the unprivileged domain.

For more details, see the *Arm® v8-M Architecture Reference Manual and Arm® Platform Security Architecture Trusted Base System Architecture for Arm®v8-M 1.0*.

### 40.3.3 Security Attribution

The TrustZone for Armv8.1-M implementation consists of the Security Attribution Unit (SAU) and Implementation Defined Attribution Unit (IDAU). The 4GB memory space is partitioned into Secure (S) and Non-secure (NS) memory regions. The Secure memory space is further divided into two types, Non-secure Callable (NSC) and Secure.

- Note:
- S = Secure addresses are used for memory and peripherals that are only accessible by secure software or secure masters.
  - NSC = A special type of secure location. This type of memory is the only type in which an Armv8.1-M processor permits to hold a Secure Gateway (SG) instruction that enables software to transition from Non-secure to Secure state.
  - NS = Non-secure addresses are used for memory and peripherals accessible by all software running on the device.

#### 40.3.3.1 Implementation Defined Attribution Unit (IDAU)

The IDAU defines the code, SRAM and peripheral region into the secure alias region and the non-secure alias region by the address bit [28]. The secure code region and secure SRAM region are assigned the NSC security attributes. The security map defined by IDAU is fixed in hardware and cannot be changed by software.

#### 40.3.3.2 Master Security Attribution Unit (MSAU)

MSAU is IDAU that defines system-specific security address map for masters other than CPU. The MSAU defines secure and non-secure alias regions but does not define the Non-secure Callable (NSC) and region number. Masters other than the CPU can issue a security transaction using the secure alias address defined by MSAU. However, non-secure masters are prohibited from issuing secure transactions using an address in the secure alias region. [Figure 40.2](#) shows the defined security map of IDAU and MSAU. The security map defined by MSAU is fixed in hardware and cannot be changed by software.



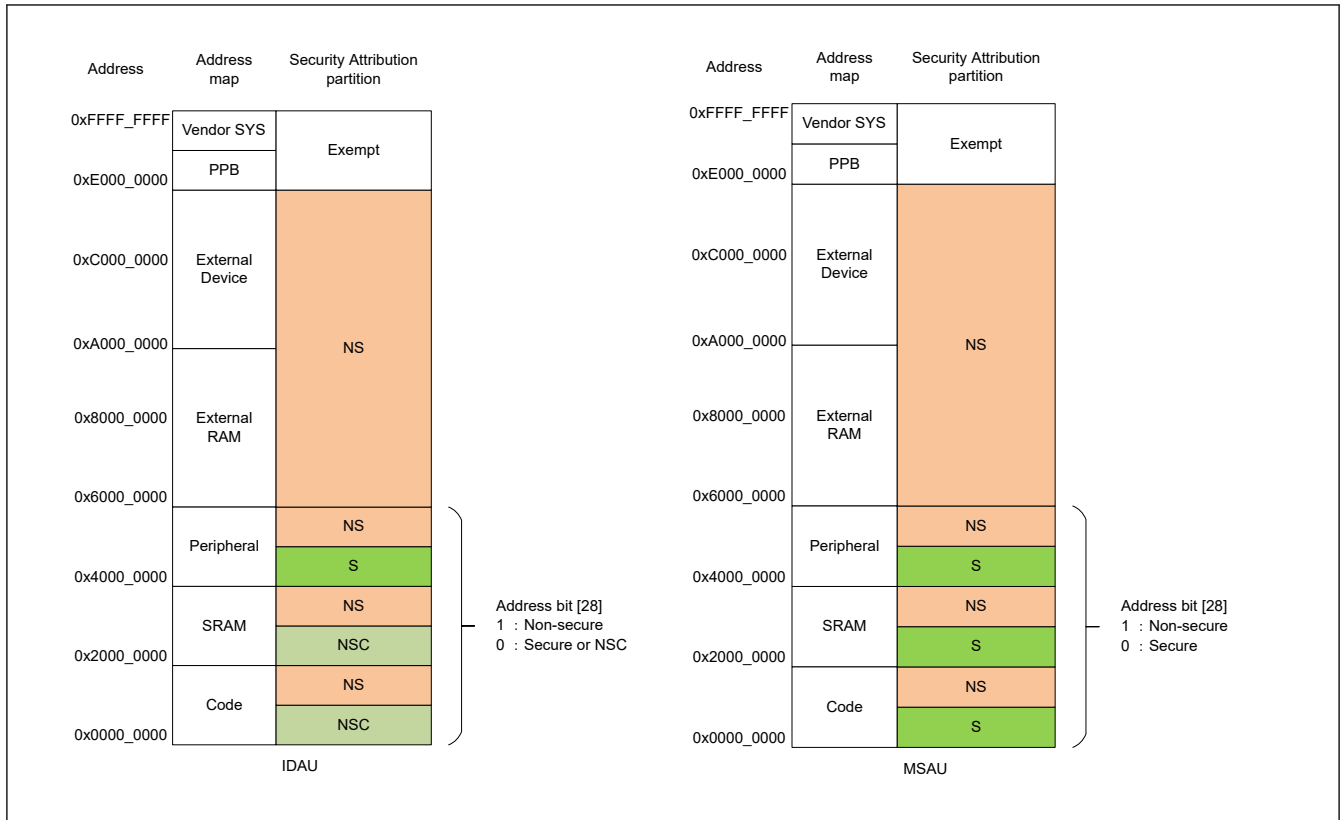


Figure 40.2 IDAU and MSAU defined security map

### 40.3.3.3 Secure Attribution Unit (SAU)

The SAU is a programmable unit that determines the security of an address. The SAU is programmable in the Secure state and has a programmers' model similar to the MPU. If an address maps to regions defined by both IDAU and SAU, the region of the highest security level is selected. Secure master can issue secure and non-secure transactions using the address of each security alias region. Non-secure master cannot issue secure transactions using the address of the secure alias region. Figure 40.3 shows the final determination of the address map security level.

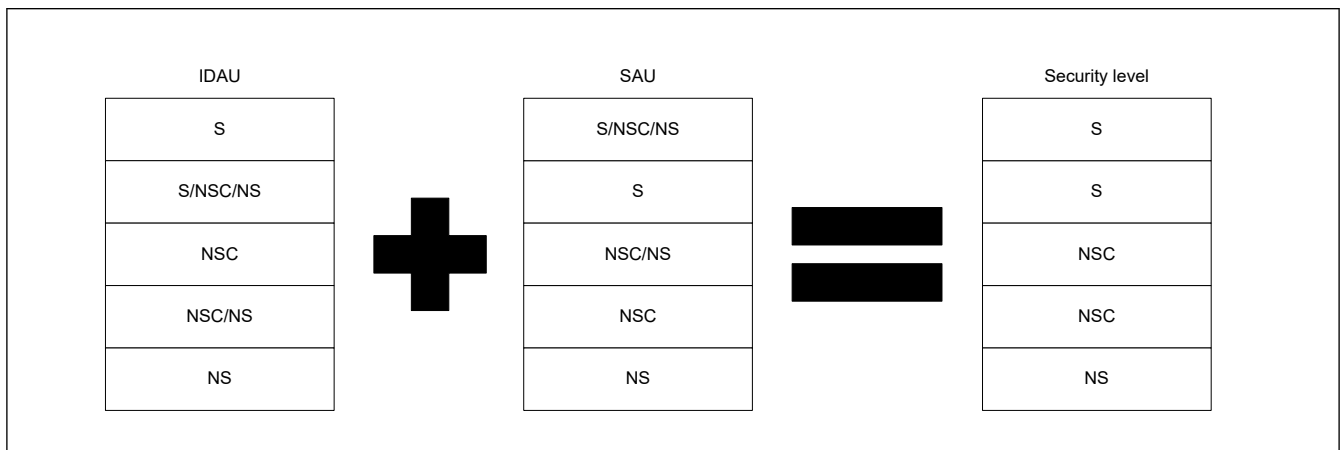


Figure 40.3 Final determination of address map security level

When using TrustZone to perform secure and non-secure region separation, SAU MUST be set according to the following.

The regions set as NS attribute in IDAU MUST be set to NS in the SAU as well. The regions set to NS attribute in IDAU are:

0x1000\_0000 to 0x1FFF\_FFFF (SAU Region 1 in Figure 40.4)

0x3000\_0000 to 0x3FFF\_FFFF (SAU Region 3 in Figure 40.4)

0x5000\_0000 to 0xDFFF\_FFFF (SAU Region 3 in Figure 40.4)

At least one NSC region MUST be created within any region defined as NSC by the IDAU. The regions set to NSC attribute by the IDAU are:

0x0000\_0000 to 0x0FFF\_FFFF (SAU Region 0 in Figure 40.4)

0x2000\_0000 to 0x2FFF\_FFFF (SAU Region 2 in Figure 40.4)

If you do not wish to define any isolation by means of using TrustZone, do not change SAU\_CTRL.ALLNS = 0 and SAU\_CTRL.ENABLE = 0 (initial value).

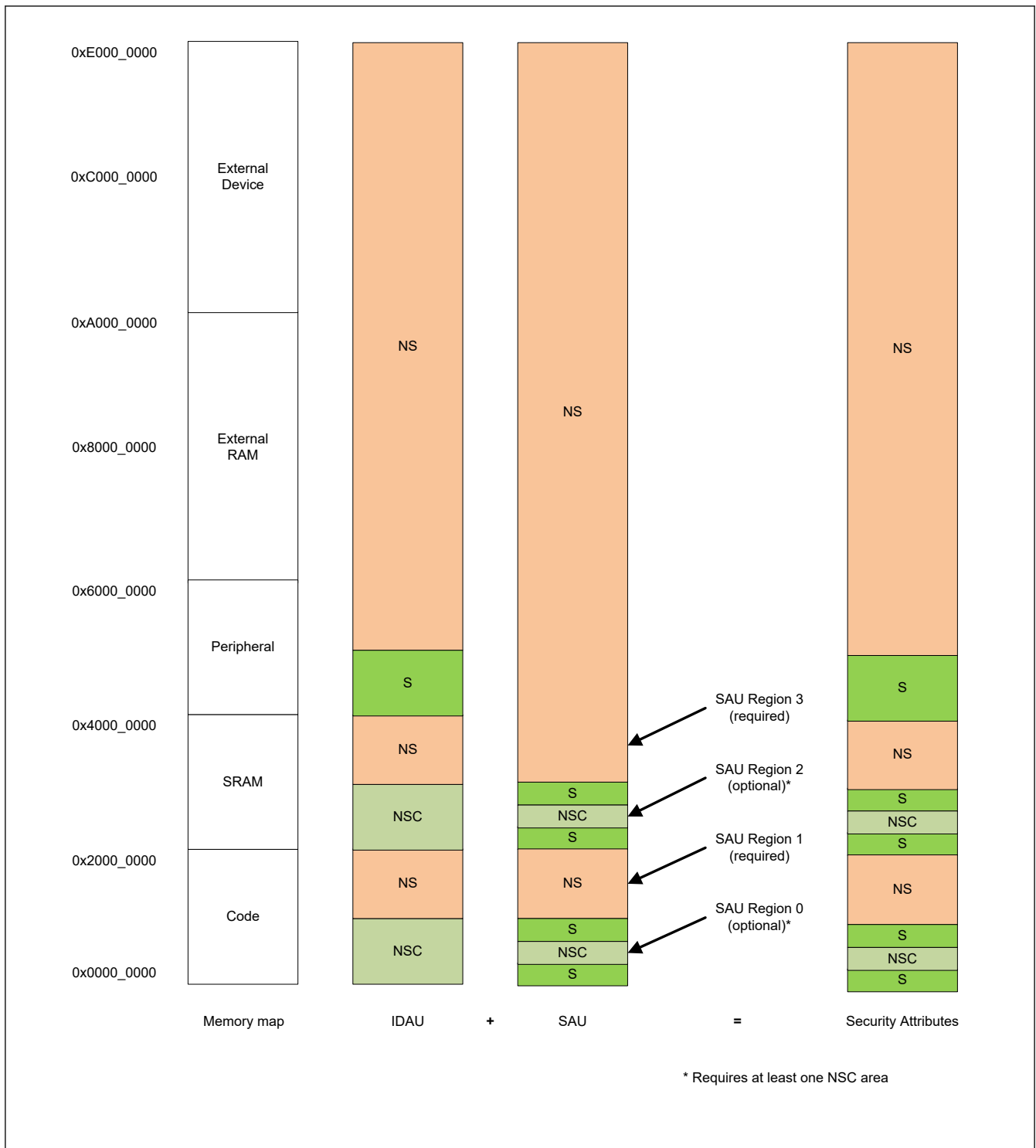


Figure 40.4 SAU settings and resulting security attributes

#### 40.3.3.4 Region Number

The SAU and IDAU also define region numbers for each of the memory regions and security attributes. This region number is used by software to determine if a contiguous range of memory shares common security attributes. [Figure 40.5](#) shows the defined region number of IDAU.

Address	Address map	Security Attribution partition	Region number
0xFFFF_FFFF	Vendor SYS	Exempt	0
0xE000_0000	PPB		
0xC000_0000	External Device	NS	6
0xA000_0000			
0x8000_0000	External RAM	NS	5
0x6000_0000			
0x4000_0000	Peripheral	S	5
0x2000_0000	SRAM	NS	4
		NSC	3
0x0000_0000	Code	NS	2
		NSC	1

**Figure 40.5** IDAU defined region numbers

#### 40.3.3.5 Memory Security Attribution of TrustZone filter

The memories are divided into S and NS regions. The memory security attribution of code flash and data flash is stored into nonvolatile memory by a boot firmware command when the device lifecycle is in the OEM state and the authentication level is AL2. These memory security attributions are applied before application execution. They cannot be updated by the application but are readable using dedicated registers. The memory security attributions of SRAM, standby SRAM, and VBATT backup registers are set by a dedicated security attribution register writable only by secure access.

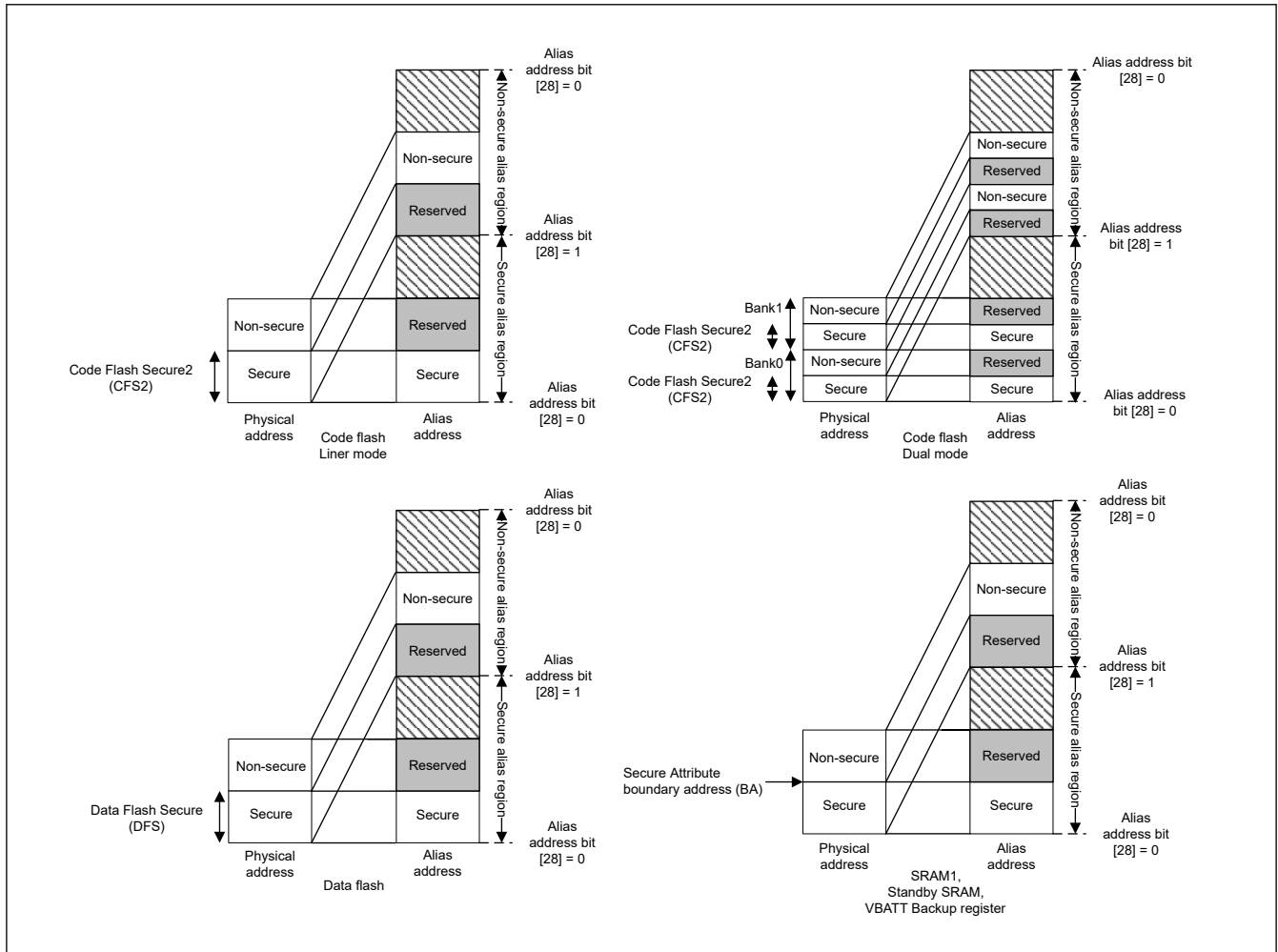
The code flash can be divided in up to two regions in linear mode and four regions in dual mode. The partitioning is the same between bank0 and bank1 in the dual mode. The data flash can be divided in up to two regions. SRAM, standby SRAM, and VBATT backup registers can be divided in up to two regions. [Figure 40.6](#) shows the memory mapping. [Table 40.1](#) shows size of the memory region and [Table 40.2](#) shows access permission of the memory region.

It is prohibited to place Secure or Non-secure Callable regions in a block swappable area in linear mode because the secure application would be placed in the non-secure region after block swapping.

Bank swapping by the Non-secure user may change the code in the Secure or Non-secure callable areas. The following is recommended to prevent this.

- Set BANKSEL\_SEL.BANKSWP[2:0] = 000b to disable the bank swapping for Non-secure users.
- Make the code in Secure or Non-secure callable areas for both banks the same when allow Non-secure user to swap banks.

See [section 40.7. Field Updating in Dual Mode.](#)



**Figure 40.6** Memory mapping

**Table 40.1** Size of memory region (1 of 2)

Memory region		Start address	Size
Linear mode	Code flash secure	0x0200_0000	CFS2 size (32 KB)
	Code flash non-secure	0x1200_0000 + CFS2 size (32 KB)	Code flash size - CFS2 size (32 KB)
Dual mode	Code flash bank0 secure	0x0200_0000	CFS2 size (32 KB)
	Code flash bank0 non-secure	0x1200_0000 + CFS2 size (32 KB)	Code flash size/2 - CFS2 size (32 KB)
	Code flash bank1 secure	0x0220_0000	CFS2 size (32 KB)
	Code flash bank1 non-secure	0x1220_0000 + CFS2 size (32 KB)	Code flash size/2 - CFS2 size (32 KB)
Data flash secure		0x2700_0000	DFS size (1 KB)
Data flash non-secure		0x3700_0000 + DFS size (1 KB)	Data flash size - DFS size (1 KB)
SRAM1 secure		0x2206_0000	BA (8 KB)
SRAM1 non-secure		0x3206_0000 + BA (8 KB)	SRAM1 size - BA (8 KB)

**Table 40.1** Size of memory region (2 of 2)

Memory region	Start address	Size
Standby SRAM secure	0x2600_0000	BA (128 bytes)
Standby SRAM non-secure	0x3600_0000 + BA (128 bytes)	Standby SRAM size - BA (128 bytes)
VBATT backup register secure	0x4001_ED00	BA (32 bytes)
VBATT backup register non-secure	0x5001_ED00 + BA (32 bytes)	Backup register size - BA (32 bytes)

Note: The number in parentheses indicates the smallest unit that can be set by the user.

Note: BA is the setting value of the security attribution boundary address register for each memory region.

**Table 40.2** Access permission of the memory region

Memory region	Secure transaction	Non-secure transaction
Each memory region configured as S or NSC	Allowed	Write ignored/Read ignored TrustZone access error is generated
Each memory region configured as NS	Write ignored/Read ignored TrustZone access error is generated	Allowed

#### 40.3.3.6 Peripheral Security Attribution of TrustZone filter

Each peripheral can be configured as secure or non-secure. Peripherals are divided into two types.

Type1 peripheral has one security attribution and access to all registers is controlled by one security attribution. Type1 peripheral security attribution is set to the PSAR<sub>x</sub> (x = B to E) register by the secure application.

Type2 peripheral has the security attribution for each register or for each bit and access to each register or bit field is controlled according to these security attributions. Type2 peripheral security attribution is set to the Security Attribution register in each module by the secure application. For details on the Security Attribution register, see each section.

Table 40.3 shows the classification of peripheral type.

**Table 40.3** Classification of peripheral type

Type1	Type2
SCI, SPI, OSPI, ETHERC, EDMAC, USBFS, IIC, RSIP-E51A, CANFD, CEU, DOC, SSIE, CRC, CAC, ACPHPS, TSN, ADC12, DAC12, POEG, AGT, GPT, ULPT, RTC, IWDT, and WDT	System control (Resets, PVD, Clock Generation Circuit, Low Power Modes, Battery Backup Function), Flash memory controller, Flash cache, SRAM controller, CPU cache, DMAC, DTC, ICU, MPU, BUS, Security setting, ELC, and I/O ports

Table 40.4 shows the access permission of type1 peripherals. The access permission of type2 peripherals is different by peripherals. See the register description of each peripherals for details.

**Table 40.4** Access permission of type1 peripherals

	Secure access	Non-secure access
Peripheral configured as S	Allowed	Write ignored/Read ignored TrustZone Access error is generated
Peripheral configured as NS	Write ignored/Read ignored TrustZone access error is generated	Allowed

#### 40.3.4 TrustZone Access Error

Table 40.5 shows the behavior when TrustZone access error occurs. The behavior varies depending on the master or the slave area to be accessed.

**Table 40.5 Behavior of TrustZone access error**

DAP	CPU	DMAC/DTC	EDMAC	CEU
<ul style="list-style-type: none"> <li>Only error response is returned<sup>*1</sup></li> </ul>	<ul style="list-style-type: none"> <li>IDAU/SAU detects SecureFault exception</li> <li>TrustZone filter detects BusFault exception</li> <li>Can issue a reset<sup>*2</sup></li> </ul>	<ul style="list-style-type: none"> <li>Stop transfer</li> <li>Issue NMI or reset<sup>*2</sup></li> <li>Generate interrupt (DMA0_TRANSERR)</li> </ul>	<ul style="list-style-type: none"> <li>Issue NMI or reset<sup>*2</sup></li> <li>Generate interrupt (ETHER_EINT0)<sup>*3</sup></li> </ul>	<ul style="list-style-type: none"> <li>Issue NMI or reset<sup>*2</sup></li> </ul>

Note: This behavior is not applicable for bufferable write access. For more information on when a bufferable write access error is detected, see [section 14.4.2. Operations When a Bus Error Occurs](#).

Note 1. When a TrustZone access error occurs by a debugger access, exception, NMI, or reset does not occur. Only the error response is returned.

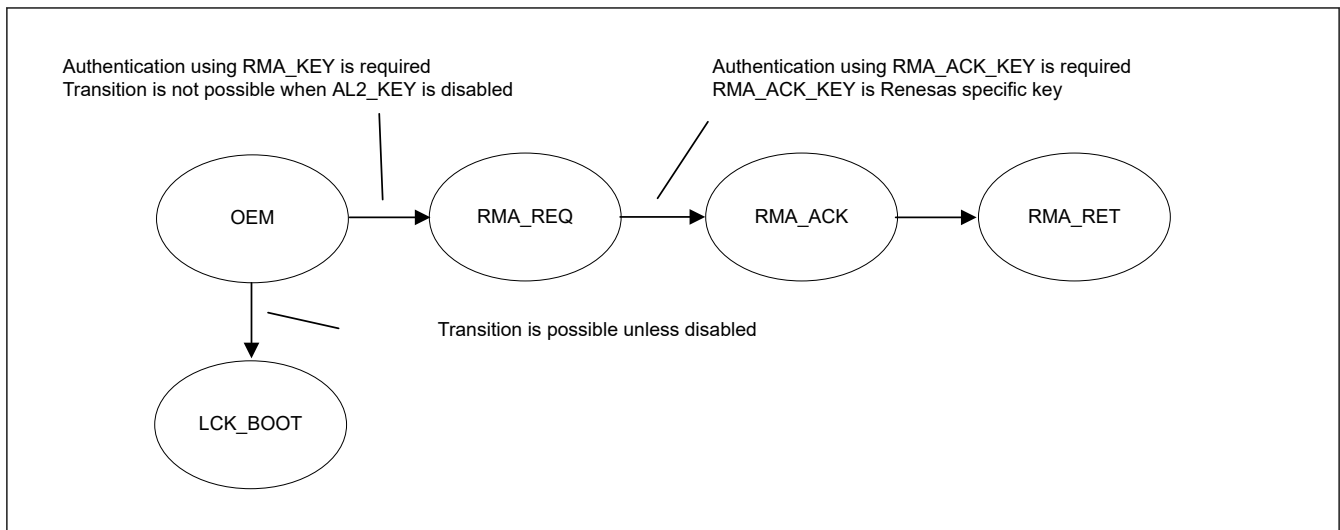
Note 2. The operation after error detection is selected by the BUSOAD.SRERROAD bit.

Note 3. Address Error flag in EESR.ADE bit is set. The interrupt occurs when it is enabled in EESIPR.ADEIP bit.

Note 4. Access error flag in STATUS.BUSIRQ is set. The interrupt occurs when it is enabled in IRQCTL.BUSIRQEN bit.

### 40.4 Device Lifecycle Management

Device Lifecycle Management (DLM) identifies the current development/production/deployment phase of the device and controls the capabilities of the debug function, the serial programming interface, and Renesas test mode. [Figure 40.7](#) shows the available device lifecycle states and [Table 40.6](#) shows the lifecycle state definitions and capabilities in each state.



**Figure 40.7 Device lifecycle states**

**Table 40.6 Lifecycle state definition and capabilities in each state (1 of 2)**

Lifecycle	Definition	Protection level	Debug function	Serial programming	Renesas test mode
OEM	“Original Equipment Manufacturer” The device is owned by the customer. The customer receives the device in this state.	PL2 or PL1 or PL0	Depend on the authentication level		Not available
LCK_BOOT	“LoCKed BOOT interface” The debug interface and the serial programming interface are permanently disabled.	PL0	Not available	Not available	Not available
RMA_REQ	“Return Material Authorization REQUEST” Request for RMA. The customer must send the device to Renesas in this state.	PL0	Not available	Available Cannot access code/ data flash area	Not available
RMA_ACK	“Return Material Authorization ACKnowledged” Failure analysis in Renesas	PL2	Available in the secure and non-secure debug	Available Cannot access code/ data flash area	Available

**Table 40.6 Lifecycle state definition and capabilities in each state (2 of 2)**

Lifecycle	Definition	Protection level	Debug function	Serial programming	Renesas test mode
RMA_RET	"Return Material Authorization RETurn" The device is back to the customer. The device does not boot.	PL0	Not available	Not available	Not available

#### 40.4.1 Changing the Lifecycle State

Use the boot firmware commands to change the device lifecycle state. These commands are available via the boot firmware's UART and USB interfaces, plus SWD/JTAG. See the boot firmware application note for details of the command. The lifecycle state cannot be updated by an application, but the current lifecycle state can be read through the dedicated registers.

As shown in [Figure 40.7](#), each transition is one-way and the state cannot be regressed.

Transition from OEM to RMA\_REQ requires key authentication using the RMA\_KEY which must have been previously injected by the customer. The key length of the RMA\_KEY is 128 bits. Inject the RMA\_KEY as shown in [section 40.5. Secure Key Injection](#). RMA\_KEY can be injected in AL2.

The key authentication uses a challenge and response authentication or authentication using the MCU's unique ID. The response (challenge and response authentication) or the authentication code (using the MCU's unique ID) can be calculated as follows:

Response = AES-128 CMAC (RMA\_KEY, 128-bit challenge)

Authentication code = AES-128 CMAC (RMA\_KEY, 128-bit MCU unique ID)

The contents of the flash memory except permanently locked blocks or registers are erased when transitioning to RMA\_REQ. The contents of the permanently locked blocks or registers can be read by Renesas at failure analysis. A flash block can permanently locked by setting the PBPS/PBPS\_SEC and BPS\_SEL registers to permanently disable programming and erasure of the block. The SAS register can be permanently locked by the FSPR bit, permanently disabling programming and erasure of the register. Transition to RMA\_REQ is not possible if the AL2\_KEY is disabled. The MCU does not respond after changing the device lifecycle state to RMA\_REQ. To continue to use boot firmware commands, you must enter boot mode again after a reset. See the boot firmware application note for details.

Transition from OEM to LCK\_BOOT is possible unless that transition has been explicitly disabled. Use the parameter setting command in AL2 or AL1 to prohibit the transition to LCK\_BOOT. The LCK\_BOOT transition prohibition is a permanent setting and cannot be undone. The debug interface and serial programming interface are permanently disabled in LCK\_BOOT.

#### 40.4.2 Protection and Authentication Level

The protection level (PL) and the authentication level (AL) determine the availability of the debug function and the serial programming interface. PL and AL are fixed except in the OEM state. For PL and for AL, three levels can be set for each in the OEM state. AL indicates a temporary authentication status and is initialized to PL after an MCU power-on reset. Both PL and AL can be changed only by the boot firmware and cannot be changed by the application. [Figure 40.8](#) shows the available Protection Level and Authentication Level states and transitions. [Table 40.7](#) shows the availability of the debug function and the serial programming interface in each level.

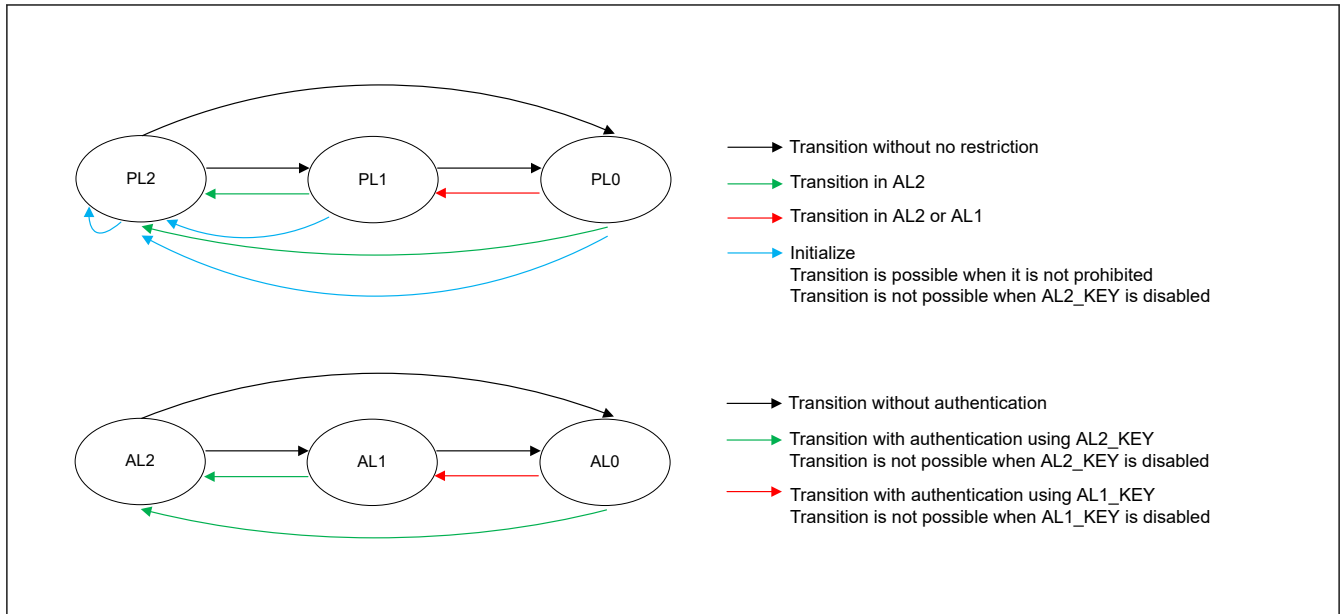


Figure 40.8 PL and AL states and transitions

Table 40.7 Availability of the debug function and the serial programming interface in each authentication level

AL	Debug function	Serial programming interface
AL2	Non-secure and Secure debug functions are enabled and accessible from the debugger	All functions are available
AL1	Only non-secure debug function is enabled, and debugger can access only defined non-secure debug accessible regions	Available but cannot program, erase, or read secure code or data flash area
AL0	No debug functions are available	Available but cannot access code or data flash area

Use the boot firmware commands to change PL and AL. See the boot firmware application note for details of the command.

Changing to a lower PL can be done with no restriction. Changing to a higher PL requires that the MCU be at that AL or higher. For example, when the current AL is AL1, changing to PL1 is possible but changing to PL2 is impossible. PL can be reset by the Initialize command unless the command itself is disabled. The Initialize command will set the PL to PL2 and the content in the flash memory will be erased. If there are any permanently locked blocks or registers, the Initialize command does not execute. In other words, when all the bits of the PBPS and PBPS\_SEC registers are 1 and the FSPR bit is 1, the Initialize command is executable. The Initialize command can be disabled permanently in all AL states by the parameter setting command to prevent users from erasing the contents of flash memory. The Initialize command is disabled also when AL2\_KEY is disabled. The MCU does not respond after executing the Initialize command. To continue to use the boot firmware commands, enter boot mode again after a reset. See the boot firmware application note for details.

Changing to a lower AL can be done without authentication. Changing to a higher AL requires key authentication using AL2\_KEY or AL1\_KEY, as appropriate. These keys are 128-bit keys. Injection of AL2\_KEY or AL1\_KEY is performed as shown in section 40.5. Secure Key Injection. AL2\_KEY can be injected in AL2, AL1\_KEY can be injected in AL2 or AL1. The key authentication uses a challenge and response authentication. The response can be calculated as follows:

$$\text{Response} = \text{AES-128 CMAC}(\text{KEY}, 128 \text{ bits challenge})$$

AL2\_KEY can be disabled permanently in AL2 by parameter setting command. AL1\_KEY can be disabled permanently in AL2 or AL1 by parameter setting command.

### 40.4.3 Serial Programming

Whether a serial programmer can be connected and the range of flash memory that can be accessed depends on the device lifecycle state and AL as shown in Table 40.6 and Table 40.7. In addition, the accepted serial programming commands differ depending on the device lifecycle state and AL. See the boot firmware application note for details of the commands.



#### 40.4.4 Device Lifecycle State and PL Change Example

The following section describes a typical device lifecycle state and PL change example.

- Secure developer
  - Set the memory security attribution of the code flash and data flash using the boot firmware command
  - Program and debug the secure application. If the memory security attribution is not set, all area of the code flash and data flash is secure.
  - If the AL2\_KEY and RMA\_KEY are required, inject them using the boot firmware commands.
  - Prepare the MCU for the Non-secure developer
    - If the Non-secure Developer will not be permitted to use the Initialize command, disable it using the boot firmware command.
    - If the AL2\_KEY will not be used and the Non-secure Developer will not be permitted to use the Initialize function, disable the AL2\_KEY using the boot firmware command.
    - Change the PL from PL2 to PL1 using the boot firmware command
- Non-secure developer
  - Program and debug the non-secure application.
  - If the AL1\_KEY is required, inject it using the boot firmware command.
  - Prepare the MCU for end-product deployment
    - Disable the Initialize command using the boot firmware command (if required)
    - If the AL1\_KEY will not be used, disable the AL1\_KEY using the boot firmware command.
    - Change the PL from PL1 to PL0 using the boot firmware command

#### 40.4.5 Failure Analysis

If the customer requests failure analysis by Renesas, it is necessary to send the device after changing the device lifecycle state to RMA\_REQ. If the device lifecycle state is not RMA\_REQ, Renesas cannot perform the failure analysis. After failure analysis, Renesas changes the lifecycle to RMA\_RET and returns the device to the customer.

Note: RMA\_KEY is required to change the device lifecycle state to RMA\_REQ or the MCU's Unique ID must be used as part of an authentication code. See [section 40.4.1. Changing the Lifecycle State](#) for details.

#### 40.5 Secure Key Injection

To inject a user key into the MCU, perform the following steps.

Renesas provides the Security Key Management Tool, available on the Renesas web site, to assist with key injection preparation.

1. Create a 256-bit installation key.  
This key is called the User Factory Programming Key (UFPK) and is used to wrap a user key
2. Get the wrapped version (W-UFPK) of the UFPK through the Renesas Key Wrapping Service.
3. Wrap the user key using the UFPK.
4. Send the W-UFPK and the wrapped user key to the MCU using a boot firmware interface.  
The user key is unwrapped, wrapped with the MCU's hardware unique key, and stored in nonvolatile memory. DLM keys are stored in unmapped flash. Application keys are stored at the address specified with the key injection command.

[Figure 40.9](#) shows an example of key injection and [Table 40.8](#) shows the keys that can be injected.

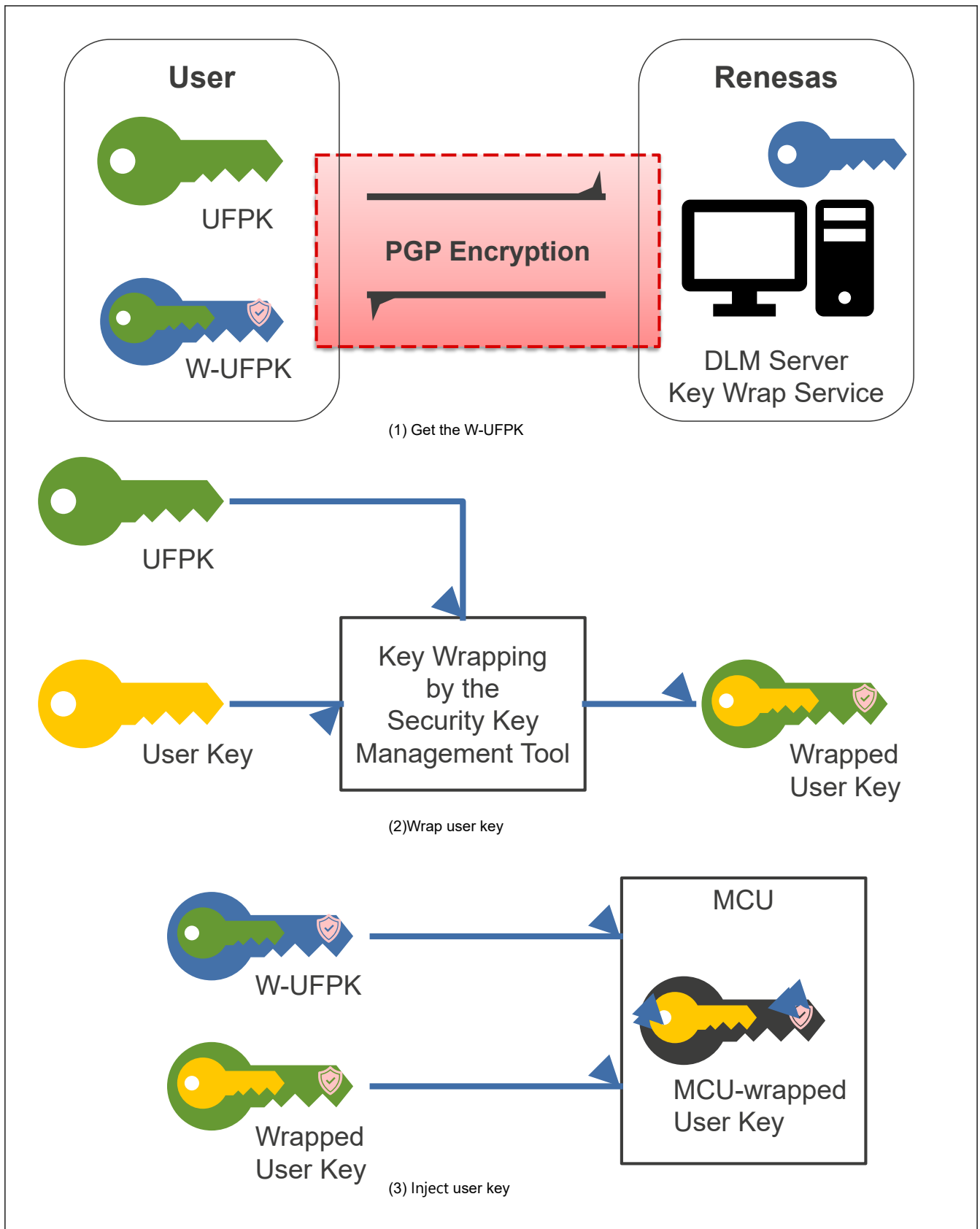


Figure 40.9 Key injection

**Table 40.8** Keys that can be injected

DLM transition	AL transition
RMA_KEY	AL2_KEY, AL1_KEY

## 40.6 Secure Factory Programming

In addition to secure key injection support for injecting user keys (DLM and debug authentication keys), the MCU supports the programming of a firmware image in ciphertext format to prevent assets from being leaked during production programming. This enables secure factory programming in a non-secure environment. Secure factory programming is supported by the boot firmware. [Figure 40.10](#) shows an example of secure factory programming of an encrypted firmware image. The customer wraps the Image Encryption Key with the UFPK and encrypts the image with the Image Encryption Key using AES128-CCM. When the customer sends the W-UFPK, the wrapped Image Encryption Key, and encrypted image to the MCU through the serial programming interface, the MCU decrypts and programs the firmware image.

To further support secure factory programming in a non-secure environment, the DLM state, Protection Level, and Authentication Keys can all be set via a single boot firmware command. Points to note about this boot firmware command include:

- Encrypted firmware programming can be performed only when the MCU is in the OEM state.
- This command will change the Protection Level of the MCU. The initial PL must be PL2. The final PL must be PL0.
- If the DLM state will remain in the OEM state, AL2 key must be injected. Do not inject AL1 key.
- The MCU can be transitioned to the LCK\_BOOT state. In this case, AL keys cannot be injected.
- AL2 key must be wrapped with the same UFPK as the Image Encryption Key.
- This command will erase all code and data flash area except option-setting memory before programming encrypted firmware image. If there are any permanently locked blocks, this command can not be executed.
- Prohibit to execute this command if the current register setting or write value to the registers related to startup area selection and startup bank selection are other than the following.
  - SAS.BTFLG = 1b
  - BANKSEL.BANKSWP[2:0] = 111b
  - BANKSEL\_SEC.BANKSWP[2:0] = 111b
- All option-setting memory values must be included in the encrypted firmware image, including settings that are not used with default values. However, if the following areas are write-protected, the write data in these areas should not be included in the image. If included, this command will terminate with an error.
  - SAS register
  - Lockable Area 0 to 2 in the data flash option setting memory

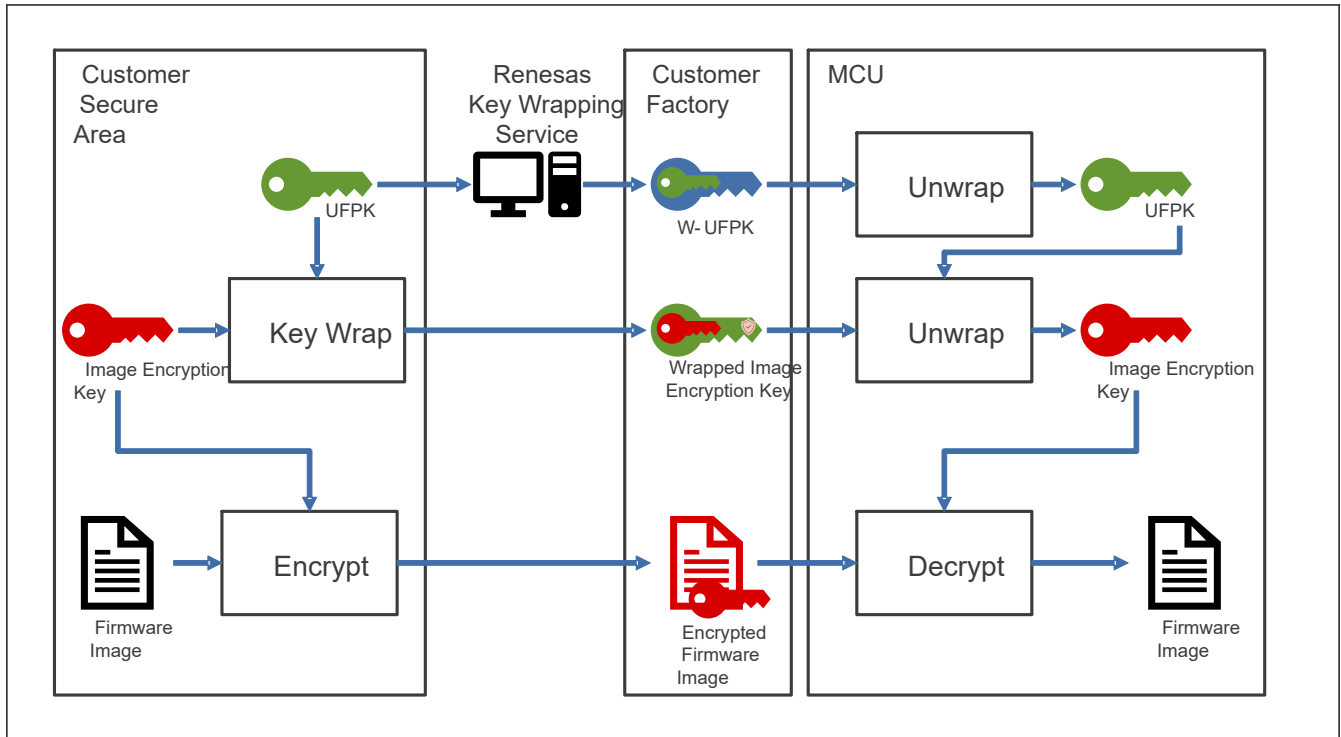


Figure 40.10 Secure factory encrypted image programming

## 40.7 Field Updating in Dual Mode

Be careful when performing in-field firmware update to keep the contents of secure (S or NSC) or Non-secure (NS) regions after bank swapping in the dual mode. Figure 40.11 and Figure 40.12 show the performing in-field update of information flow of secure or non-secure regions in the dual mode.

Figure 40.11 and Figure 40.12 show updating the Secure and the Non-secure regions respectively when operating in dual bank mode. Initially, BANK0 contains Version 1 of the firmware, and BANK1 is blank. As shown in Figure 40.11, to update the S or NSC firmware to Version 2, first copy the Version 1 NS firmware in BANK0 to BANK1, update the S or NSC to Version 2, and then execute the bank swap.

As shown in Figure 40.12, to update the NS firmware to Version 2, first copy the Version 1 S and NSC firmware from BANK0 to BANK1. Note that this process must be done by a secure service. After that, non-secure users should update the NS firmware to Version 2 and execute the bank swap.

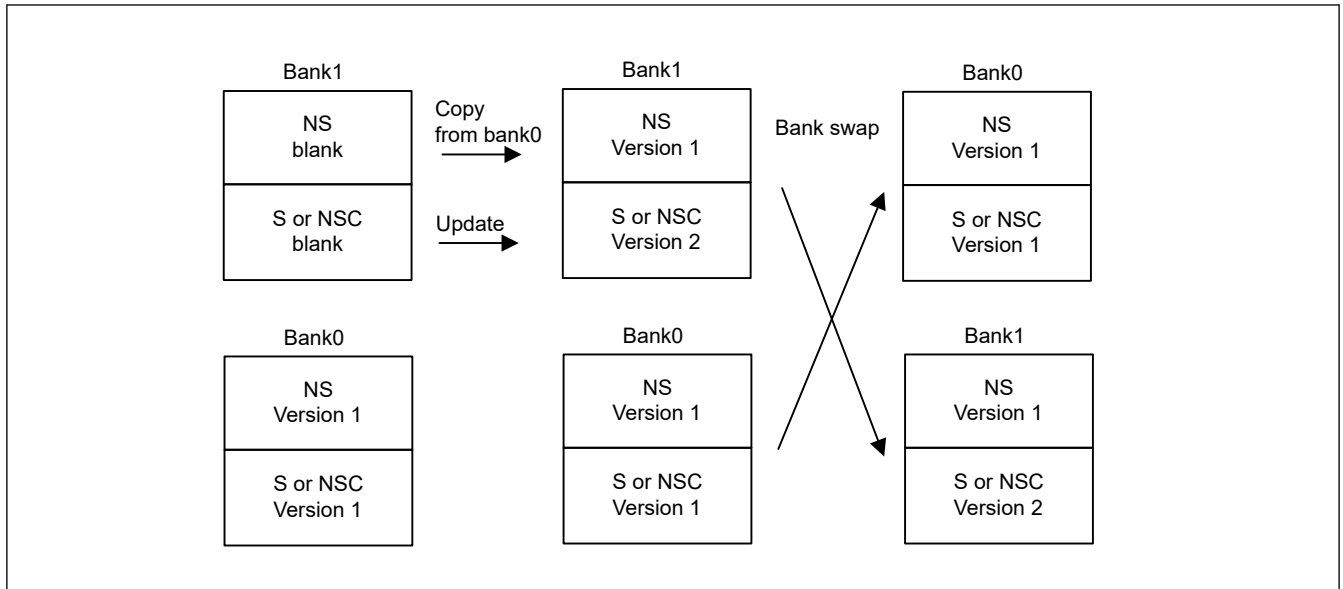
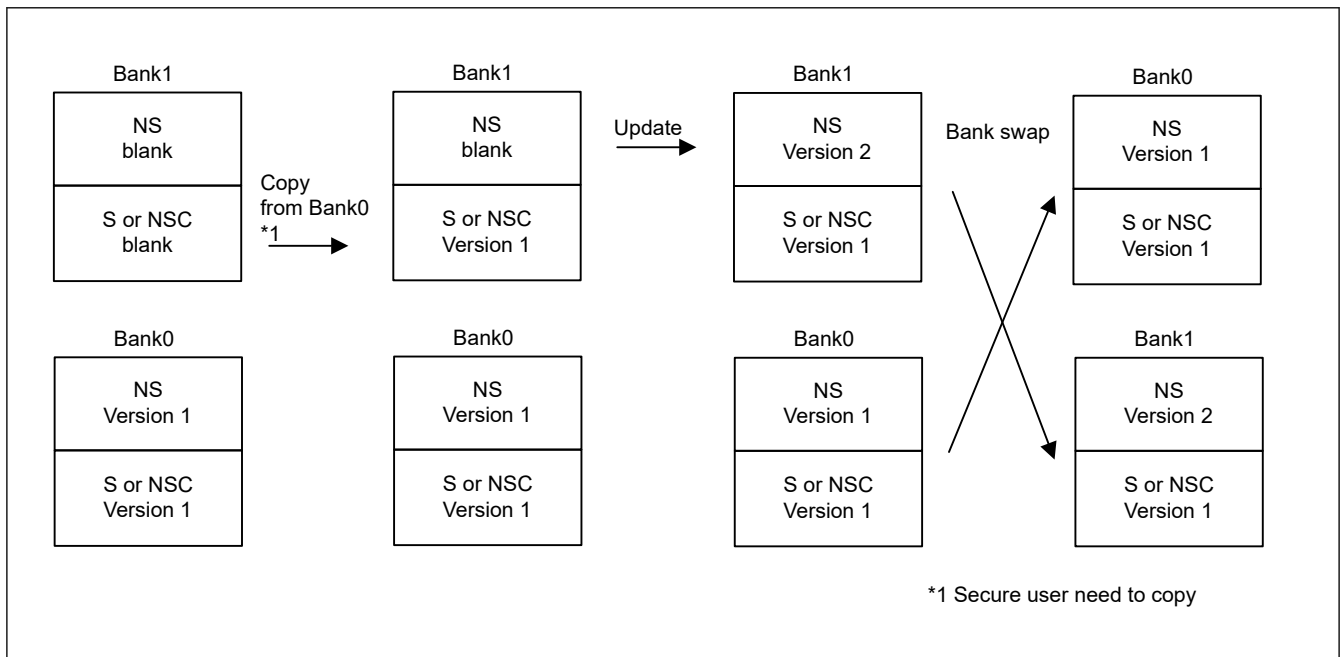


Figure 40.11 Field updating flow of secure or non-secure callable regions



\*1 Secure user need to copy

Figure 40.12 Field updating flow of non-secure region

## 40.8 Register Description

### 40.8.1 PSARB : Peripheral Security Attribution Register B

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PSAR B31	PSAR B30	PSAR B29	PSAR B28	PSAR B27	—	—	—	—	PSAR B22	—	—	PSAR B19	PSAR B18	—	PSAR B16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSAR B15	—	—	—	PSAR B11	—	PSAR B9	PSAR B8	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	PSARB8	I2C Bus Interface 1 Security Attribution Target module: IIC1 and the MSTPCRB.MSTPB8 bit 0: Secure 1: Non-secure	R/W
9	PSARB9	I2C Bus Interface 0 Security Attribution Target module: IIC0 and the MSTPCRB.MSTPB9 bit 0: Secure 1: Non-secure	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	PSARB11	Universal Serial Bus 2.0 FS Interface 0 Security Attribution Target module: USBFS0 and the MSTPCRB.MSTPB11 bit 0: Secure 1: Non-secure	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	PSARB15	ETHERC/EDMAC Controller Security Attribution Target module: ETHERC/EDMAC, the MSTPCRB.MSTPB15 bit, and the PFENET.PHYMODE0 bit 0: Secure 1: Non-secure	R/W
16	PSARB16	Octa Memory Controller Security Attribution Target module: OSPI and the MSTPCRB.MSTPB16 bit 0: Secure 1: Non-secure	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W
18	PSARB18	Serial Peripheral Interface 1 Security Attribution Target module: RSPI1 and the MSTPCRB.MSTPB18 bit 0: Secure 1: Non-secure	R/W
19	PSARB19	Serial Peripheral Interface 0 Security Attribution Target module: RSPI0 and the MSTPCRB.MSTPB19 bit 0: Secure 1: Non-secure	R/W
21:20	—	These bits are read as 0. The write value should be 0.	R/W
22	PSARB22	Serial Communication Interface 9 Security Attribution Target module: SC19 and the MSTPCRB.MSTPB22 bit 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
26:23	—	These bits are read as 0. The write value should be 0.	R/W
27	PSARB27	Serial Communication Interface 4 Security Attribution Target module: SCI4 and the MSTPCRB.MSTPB27 bit 0: Secure 1: Non-secure	R/W
28	PSARB28	Serial Communication Interface 3 Security Attribution Target module: SCI3 and the MSTPCRB.MSTPB28 bit 0: Secure 1: Non-secure	R/W
29	PSARB29	Serial Communication Interface 2 Security Attribution Target module: SCI2 and the MSTPCRB.MSTPB29 bit 0: Secure 1: Non-secure	R/W
30	PSARB30	Serial Communication Interface 1 Security Attribution Target module: SCI1 and the MSTPCRB.MSTPB30 bit 0: Secure 1: Non-secure	R/W
31	PSARB31	Serial Communication Interface 0 Security Attribution Target module: SCI0 and the MSTPCRB.MSTPB31 bit 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1.

The PSARB register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

### 40.8.2 PSARC : Peripheral Security Attribution Register C

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PSAR C31	—	—	—	PSAR C27	PSAR C26	—	—	—	—	—	—	—	—	—	PSAR C16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	PSAR C13	—	—	—	—	PSAR C8	PSAR C7	—	—	—	—	—	PSAR C1	PSAR C0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PSARC0	Clock Frequency Accuracy Measurement Circuit Security Attribution register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register. Target module: CAC and the MSTPCRC.MSTPC0 bit 0: Secure 1: Non-secure	R/W
1	PSARC1	Cyclic Redundancy Check Calculator Security Attribution Target module: CRC and the MSTPCRC.MSTPC1 bit 0: Secure 1: Non-secure	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	PSARC7	Serial Sound Interface Enhanced (channel 1) Security Attribution Target module: SSIE1 and the MSTPCRC.MSTPC7 bit 0: Secure 1: Non-secure	R/W

Bit	Symbol	Function	R/W
8	PSARC8	Serial Sound Interface Enhanced (channel 0) Security Attribution Target module: SSIE0 and the MSTPCRC.MSTPC8 bit 0: Secure 1: Non-secure	R/W
12:9	—	These bits are read as 0. The write value should be 0.	R/W
13	PSARC13	Data Operation Circuit Security Attribution Target module: DOC and the MSTPCRC.MSTPC13 bit 0: Secure 1: Non-secure	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	PSARC16	CEU Security Attribution Target module: CEU and the MSTPCRC.MSTPC16 bit 0: Secure 1: Non-secure	R/W
25:17	—	These bits are read as 0. The write value should be 0.	R/W
26	PSARC26	Controller Area Network with Flexible Data-Rate 1 Security Attribution Target module: CANFD1 and the MSTPCRC.MSTPC26 bit 0: Secure 1: Non-secure	R/W
27	PSARC27	Controller Area Network with Flexible Data-Rate 0 Security Attribution Target module: CANFD0 and the MSTPCRC.MSTPC27 bit 0: Secure 1: Non-secure	R/W
30:28	—	These bits are read as 0. The write value should be 0.	R/W
31	PSARC31	RSIP-E51A Security Attribution Target module:RSIP-E51A and the MSTPCRC.MSTPC31 bit 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1.

The PSARC register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

### 40.8.3 PSARD : Peripheral Security Attribution Register D

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	PSAR D28	PSAR D27	—	—	—	—	PSAR D22	—	PSAR D20	—	—	—	PSAR D16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSAR D15	PSAR D14	PSAR D13	PSAR D12	PSAR D11	—	—	—	—	—	PSAR D5	PSAR D4	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	PSARD4	Asynchronous General Purpose Timer 1 Security Attribution Target module: AGT1 and the MSTPCRD.MSTPD4 bit 0: Secure 1: Non-secure	R/W



Bit	Symbol	Function	R/W
5	PSARD5	Asynchronous General Purpose Timer 0 Security Attribution Target module: AGT0 and the MSTPCRD.MSTPD5 bit 0: Secure 1: Non-secure	R/W
10:6	—	These bits are read as 0. The write value should be 0.	R/W
11	PSARD11	Port Output Enable for GPT Group 3 Security Attribution Target module: POEG Group D and the MSTPCRD.MSTPD11 bit 0: Secure 1: Non-secure	R/W
12	PSARD12	Port Output Enable for GPT Group 2 Security Attribution Target module: POEG Group C and the MSTPCRD.MSTPD12 bit 0: Secure 1: Non-secure	R/W
13	PSARD13	Port Output Enable for GPT Group 1 Security Attribution Target module: POEG Group B and the MSTPCRD.MSTPD13 bit 0: Secure 1: Non-secure	R/W
14	PSARD14	Port Output Enable for GPT Group 0 Security Attribution Target module: POEG Group A and the MSTPCRD.MSTPD14 bit 0: Secure 1: Non-secure	R/W
15	PSARD15	12-Bit A/D 1 Converter Security Attribution Target module: ADC12_1 and the MSTPCRD.MSTPD15 bit 0: Secure 1: Non-secure	R/W
16	PSARD16	12-Bit A/D 0 Converter Security Attribution Target module: ADC12_0 and the MSTPCRD.MSTPD16 bit 0: Secure 1: Non-secure	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	PSARD20	12-Bit D/A Converter Security Attribution Target module: DAC12 and the MSTPCRD.MSTPD20 bit 0: Secure 1: Non-secure	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	PSARD22	Temperature Sensor Security Attribution Target module: TSN and the MSTPCRD.MSTPD22 bit 0: Secure 1: Non-secure	R/W
26:23	—	These bits are read as 0. The write value should be 0.	R/W
27	PSARD27	High Speed Analog Comparator 1 Security Attribution Target module: ACMPHS1 and the MSTPCRD.MSTPD27 bit 0: Secure 1: Non-secure	R/W
28	PSARD28	High Speed Analog Comparator 0 Security Attribution Target module: ACMPHS0 and the MSTPCRD.MSTPD28 bit 0: Secure 1: Non-secure	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1.

The PSARD register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

#### 40.8.4 PSARE : Peripheral Security Attribution Register E

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PSAR E31	PSAR E30	PSAR E29	PSAR E28	PSAR E27	PSAR E26	—	—	—	—	PSAR E21	PSAR E20	PSAR E19	PSAR E18	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PSAR E9	PSAR E8	—	—	—	—	PSAR E3	PSAR E2	PSAR E1	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	PSARE1	WDT Security Attribution Target module: WDT 0: Secure 1: Non-secure	R/W
2	PSARE2	IWDT Security Attribution Target module: IWDT 0: Secure 1: Non-secure	R/W
3	PSARE3	Real Time Clock Security Attribution Target module: RTC 0: Secure 1: Non-secure	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	PSARE8	ULPT1 Security Attribution Target module: ULPT1 and the MSTPCRE.MSTPE8 bit 0: Secure 1: Non-secure	R/W
9	PSARE9	ULPT0 Security Attribution Target module: ULPT0 and the MSTPCRE.MSTPE9 bit 0: Secure 1: Non-secure	R/W
17:10	—	These bits are read as 0. The write value should be 0.	R/W
18	PSARE18	General PWM Timer Channel 13 Security Attribution Target module: GPT13 and the MSTPCRE.MSTPE18 bit 0: Secure 1: Non-secure	R/W
19	PSARE19	General PWM Timer Channel 12 Security Attribution Target module: GPT12 and the MSTPCRE.MSTPE19 bit 0: Secure 1: Non-secure	R/W
20	PSARE20	General PWM Timer Channel 11 Security Attribution Target module: GPT11 and the MSTPCRE.MSTPE20 bit 0: Secure 1: Non-secure	R/W
21	PSARE21	General PWM Timer Channel 10 Security Attribution Target module: GPT10 and the MSTPCRE.MSTPE21 bit 0: Secure 1: Non-secure	R/W
25:22	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
26	PSARE26	General PWM Timer Channel 5 Security Attribution Target module: GPT5 and the MSTPCRE.MSTPE26 bit 0: Secure 1: Non-secure	R/W
27	PSARE27	General PWM Timer Channel 4 Security Attribution Target module: GPT4 and the MSTPCRE.MSTPE27 bit 0: Secure 1: Non-secure	R/W
28	PSARE28	General PWM Timer Channel 3 Security Attribution Target module: GPT3 and the MSTPCRE.MSTPE28 bit 0: Secure 1: Non-secure	R/W
29	PSARE29	General PWM Timer Channel 2 Security Attribution Target module: GPT2 and the MSTPCRE.MSTPE29 bit 0: Secure 1: Non-secure	R/W
30	PSARE30	General PWM Timer Channel 1 Security Attribution Target module: GPT1 and the MSTPCRE.MSTPE30 bit 0: Secure 1: Non-secure	R/W
31	PSARE31	General PWM Timer Channel 0 Security Attribution Target module: GPT0 and the MSTPCRE.MSTPE31 bit 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1.

The PSARE register specifies the security attribution for each module and the corresponding bit in Module Stop Control Register.

### 40.8.5 MSSAR : Module Stop Security Attribution Register

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MSSA R31	—	—	—	—	—	—	—	—	MSSA R22	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MSSA R15	—	—	—	—	—	—	—	—	—	—	—	—	—	MSSA R1	MSSA R0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MSSAR0	Unnecessary Circuit Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA0 bit 0: Secure 1: Non-secure	R/W
1	MSSAR1	SRAM1 Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA1 bit 0: Secure 1: Non-secure	R/W
14:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15	MSSAR15	Standby RAM Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA15 bit 0: Secure 1: Non-secure	R/W
21:16	—	These bits are read as 0. The write value should be 0.	R/W
22	MSSAR22	DMAC/DTC Clock Stop Security Attribution Target module: the MSTPCRA.MSTPA22 bit 0: Secure 1: Non-secure	R/W
30:23	—	These bits are read as 0. The write value should be 0.	R/W
31	MSSAR31	ELC Clock Stop Security Attribution Target module: the MSTPCRC.MSTPC14 bit 0: Secure 1: Non-secure	R/W

Note: S-TYPE-1, P-TYPE-1.

The MSSAR register specifies the security attribution for the corresponding bit in Module Stop Control Register.

### 40.8.6 PPARB : Peripheral Privilege Attribution Register B

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PPAR B31	PPAR B30	PPAR B29	PPAR B28	PPAR B27	—	—	—	—	PPAR B22	—	—	PPAR B19	PPAR B18	—	PPAR B16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PPAR B15	—	—	—	PPAR B11	—	PPAR B9	PPAR B8	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 1. The write value should be 1.	R/W
8	PPARB8	I2C Bus Interface 1 Privilege Attribution Target module: IIC1 and the MSTPCRB.MSTPB8 bit 0: Privileged 1: UnPrivileged	R/W
9	PPARB9	I2C Bus Interface 0 Privilege Attribution Target module: IIC0 and the MSTPCRB.MSTPB9 bit 0: Privileged 1: UnPrivileged	R/W
10	—	This bit is read as 1. The write value should be 1.	R/W
11	PPARB11	Universal Serial Bus 2.0 FS Interface 0 Privilege Attribution Target module: USBFS0 and the MSTPCRB.MSTPB11 bit 0: Privileged 1: UnPrivileged	R/W
14:12	—	These bits are read as 1. The write value should be 1.	R/W
15	PPARB15	ETHERC/EDMAC Controller Privilege Attribution Target module: ETHERC/EDMAC, the MSTPCRB.MSTPB15 bit, and the PFENET.PHYMODE0 bit 0: Privileged 1: UnPrivileged	R/W

Bit	Symbol	Function	R/W
16	PPARB16	Octa Memory Controller Privilege Attribution Target module: OSPI and the MSTPCRB.MSTPB16 bit 0: Privileged 1: UnPrivileged	R/W
17	—	This bit is read as 1. The write value should be 1.	R/W
18	PPARB18	Serial Peripheral Interface 1 Privilege Attribution Target module: RSPI1 and the MSTPCRB.MSTPB18 bit 0: Privileged 1: UnPrivileged	R/W
19	PPARB19	Serial Peripheral Interface 0 Privilege Attribution Target module: RSPI0 and the MSTPCRB.MSTPB19 bit 0: Privileged 1: UnPrivileged	R/W
21:20	—	These bits are read as 1. The write value should be 1.	R/W
22	PPARB22	Serial Communication Interface 9 Privilege Attribution Target module: SC19 and the MSTPCRB.MSTPB22 bit 0: Privileged 1: UnPrivileged	R/W
26:23	—	These bits are read as 1. The write value should be 1.	R/W
27	PPARB27	Serial Communication Interface 4 Privilege Attribution Target module: SC14 and the MSTPCRB.MSTPB27 bit 0: Privileged 1: UnPrivileged	R/W
28	PPARB28	Serial Communication Interface 3 Privilege Attribution Target module: SC13 and the MSTPCRB.MSTPB28 bit 0: Privileged 1: UnPrivileged	R/W
29	PPARB29	Serial Communication Interface 2 Privilege Attribution Target module: SC12 and the MSTPCRB.MSTPB29 bit 0: Privileged 1: UnPrivileged	R/W
30	PPARB30	Serial Communication Interface 1 Privilege Attribution Target module: SC11 and the MSTPCRB.MSTPB30 bit 0: Privileged 1: UnPrivileged	R/W
31	PPARB31	Serial Communication Interface 0 Privilege Attribution Target module: SC10 and the MSTPCRB.MSTPB31 bit 0: Privileged 1: UnPrivileged	R/W

Note: S-TYPE-2, P-TYPE-1.

The PPARB register specifies the privilege attribution for each module and the corresponding bit in Module Stop Control Register.

### 40.8.7 PPARC : Peripheral Privilege Attribution Register C

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PPAR C31	—	—	—	PPAR C27	PPAR C26	—	—	—	—	—	—	—	—	—	PPAR C16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	PPAR C13	—	—	—	—	PPAR C8	PPAR C7	—	—	—	—	—	PPAR C1	PPAR C0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	PPARC0	Clock Frequency Accuracy Measurement Circuit Privilege Attribution Target module: CAC and the MSTPCRC.MSTPC0 bit 0: Privileged 1: UnPrivileged	R/W
1	PPARC1	Cyclic Redundancy Check Calculator Privilege Attribution Target module: CRC and the MSTPCRC.MSTPC1 bit 0: Privileged 1: UnPrivileged	R/W
6:2	—	These bits are read as 1. The write value should be 1.	R/W
7	PPARC7	Serial Sound Interface Enhanced (Channel 1) Privilege Attribution Target module: SSIE1 and the MSTPCRC.MSTPC7 bit 0: Privileged 1: UnPrivileged	R/W
8	PPARC8	Serial Sound Interface Enhanced (Channel 0) Privilege Attribution Target module: SSIE0 and the MSTPCRC.MSTPC8 bit 0: Privileged 1: UnPrivileged	R/W
12:9	—	These bits are read as 1. The write value should be 1.	R/W
13	PPARC13	Data Operation Circuit Privilege Attribution Target module: DOC and the MSTPCRC.MSTPC13 bit 0: Privileged 1: UnPrivileged	R/W
15:14	—	These bits are read as 1. The write value should be 1.	R/W
16	PPARC16	CEU Privilege Attribution Target module: CEU and the MSTPCRC.MSTPC16 bit 0: Privileged 1: UnPrivileged	R/W
25:17	—	These bits are read as 1. The write value should be 1.	R/W
26	PPARC26	Controller Area Network with Flexible Data-Rate 1 Privilege Attribution Target module: CANFD1 and the MSTPCRC.MSTPC26 bit 0: Privileged 1: UnPrivileged	R/W
27	PPARC27	Controller Area Network with Flexible Data-Rate 0 Privilege Attribution Target module: CANFD0 and the MSTPCRC.MSTPC27 bit 0: Privileged 1: UnPrivileged	R/W
30:28	—	These bits are read as 1. The write value should be 1.	R/W
31	PPARC31	RSIP-E51A Privilege Attribution Target module: RSIP-E51A and the MSTPCRC.MSTPC31 bit 0: Privileged 1: UnPrivileged	R/W

Note: S-TYPE-2, P-TYPE-1.

The PPARC register specifies the privilege attribution for each module and the corresponding bit in Module Stop Control Register.

### 40.8.8 PPARD : Peripheral Privilege Attribution Register D

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	PPAR D28	PPAR D27	—	—	—	—	PPAR D22	—	PPAR D20	—	—	—	PPAR D16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PPAR D15	PPAR D14	PPAR D13	PPAR D12	PPAR D11	—	—	—	—	—	PPAR D5	PPAR D4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 1. The write value should be 1.	R/W
4	PPARD4	Asynchronous General Purpose Timer 1 Privilege Attribution Target module: AGT1 and the MSTPCRD.MSTPD4 bit 0: Privileged 1: UnPrivileged	R/W
5	PPARD5	Asynchronous General Purpose Timer 0 Privilege Attribution Target module: AGT0 and the MSTPCRD.MSTPD5 bit 0: Privileged 1: UnPrivileged	R/W
10:6	—	These bits are read as 1. The write value should be 1.	R/W
11	PPARD11	Port Output Enable for GPT Group 3 Privilege Attribution Target module: POEG Group D and the MSTPCRD.MSTPD11 bit 0: Privileged 1: UnPrivileged	R/W
12	PPARD12	Port Output Enable for GPT Group 2 Privilege Attribution Target module: POEG Group C and the MSTPCRD.MSTPD12 bit 0: Privileged 1: UnPrivileged	R/W
13	PPARD13	Port Output Enable for GPT Group 1 Privilege Attribution Target module: POEG Group B and the MSTPCRD.MSTPD13 bit 0: Privileged 1: UnPrivileged	R/W
14	PPARD14	Port Output Enable for GPT Group 0 Privilege Attribution Target module: POEG Group A and the MSTPCRD.MSTPD14 bit 0: Privileged 1: UnPrivileged	R/W
15	PPARD15	12-Bit A/D 1 Converter Privilege Attribution Target module: ADC12_1 and the MSTPCRD.MSTPD15 bit 0: Privileged 1: UnPrivileged	R/W
16	PPARD16	12-Bit A/D 0 Converter Privilege Attribution Target module: ADC12_0 and the MSTPCRD.MSTPD16 bit 0: Privileged 1: UnPrivileged	R/W
19:17	—	These bits are read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
20	PPARD20	12-Bit D/A Converter Privilege Attribution Target module: DAC12 and the MSTPCRD.MSTPD20 bit 0: Privileged 1: UnPrivileged	R/W
21	—	This bit is read as 1. The write value should be 1.	R/W
22	PPARD22	Temperature Sensor Privilege Attribution Target module: TSN and the MSTPCRD.MSTPD22 bit 0: Privileged 1: UnPrivileged	R/W
26:23	—	These bits are read as 1. The write value should be 1.	R/W
27	PPARD27	High speed analog Comparator 1 Privilege Attribution Target module: ACMPHS1 and the MSTPCRD.MSTPD27 bit 0: Privileged 1: UnPrivileged	R/W
28	PPARD28	High speed analog Comparator 0 Privilege Attribution Target module: ACMPHS0 and the MSTPCRD.MSTPD28 bit 0: Privileged 1: UnPrivileged	R/W
31:29	—	These bits are read as 1. The write value should be 1.	R/W

Note: S-TYPE-2, P-TYPE-1.

The PPARD register specifies the privilege attribution for each module and the corresponding bit in Module Stop Control Register.

#### 40.8.9 PPARE : Peripheral Privilege Attribution Register E

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PPAR E31	PPAR E30	PPAR E29	PPAR E28	PPAR E27	PPAR E26	—	—	—	—	PPAR E21	PPAR E20	PPAR E19	PPAR E18	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PPAR E9	PPAR E8	—	—	—	—	PPAR E3	PPAR E2	PPAR E1	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	—	This bit is read as 1. The write value should be 1.	R/W
1	PPARE1	WDT Privilege Attribution Target module: WDT 0: Privileged 1: UnPrivileged	R/W
2	PPARE2	IWDT Privilege Attribution Target module: IWDT 0: Privileged 1: UnPrivileged	R/W
3	PPARE3	Real Time Clock Privilege Attribution Target module: RTC 0: Privileged 1: UnPrivileged	R/W
7:4	—	These bits are read as 1. The write value should be 1.	R/W



Bit	Symbol	Function	R/W
8	PPARE8	ULPT1 Privilege Attribution Target module: ULPT1 and the MSTPCRE.MSTPE8 bit 0: Privileged 1: UnPrivileged	R/W
9	PPARE9	ULPT0 Privilege Attribution Target module: ULPT0 and the MSTPCRE.MSTPE9 bit 0: Privileged 1: UnPrivileged	R/W
17:10	—	These bits are read as 1. The write value should be 1.	R/W
18	PPARE18	General PWM Timer Channel 13 Privilege Attribution Target module: : GPT13 and the MSTPCRE.MSTPE18 bit 0: Privileged 1: UnPrivileged	R/W
19	PPARE19	General PWM Timer Channel 12 Privilege Attribution Target module: : GPT12 and the MSTPCRE.MSTPE19 bit 0: Privileged 1: UnPrivileged	R/W
20	PPARE20	General PWM Timer Channel 11 Privilege Attribution Target module: : GPT11 and the MSTPCRE.MSTPE20 bit 0: Privileged 1: UnPrivileged	R/W
21	PPARE21	General PWM Timer Channel 10 Privilege Attribution Target module: : GPT10 and the MSTPCRE.MSTPE21 bit 0: Privileged 1: UnPrivileged	R/W
25:22	—	These bits are read as 1. The write value should be 1.	R/W
26	PPARE26	General PWM Timer Channel 5 Privilege Attribution Target module: : GPT5 and the MSTPCRE.MSTPE26 bit 0: Privileged 1: UnPrivileged	R/W
27	PPARE27	General PWM Timer Channel 4 Privilege Attribution Target module: : GPT4 and the MSTPCRE.MSTPE27 bit 0: Privileged 1: UnPrivileged	R/W
28	PPARE28	General PWM Timer Channel 3 Privilege Attribution Target module: : GPT3 and the MSTPCRE.MSTPE28 bit 0: Privileged 1: UnPrivileged	R/W
29	PPARE29	General PWM Timer Channel 2 Privilege Attribution Target module: : GPT2 and the MSTPCRE.MSTPE29 bit 0: Privileged 1: UnPrivileged	R/W
30	PPARE30	General PWM Timer Channel 1 Privilege Attribution Target module: : GPT1 and the MSTPCRE.MSTPE30 bit 0: Privileged 1: UnPrivileged	R/W
31	PPARE31	General PWM Timer Channel 0 Privilege Attribution Target module: : GPT0 and the MSTPCRE.MSTPE31 bit 0: Privileged 1: UnPrivileged	R/W

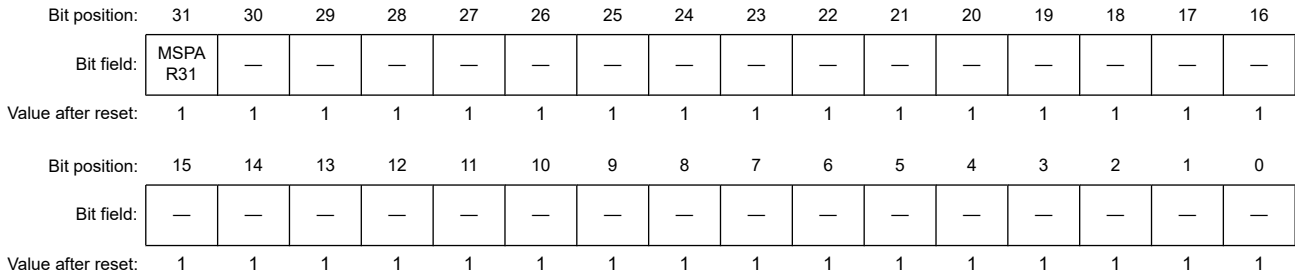
Note: S-TYPE-2, P-TYPE-1.

The PPARE register specifies the privilege attribution for each module and the corresponding bit in Module Stop Control Register.

### 40.8.10 MSPAR : Module Stop Privilege Attribution Register

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x2C



Bit	Symbol	Function	R/W
30:0	—	These bits are read as 1. The write value should be 1.	R/W
31	MSPAR31	ELC Clock Stop Privilege Attribution Target module: the MSTPCRC.MSTPC14 bit 0: Privileged 1: UnPrivileged	R/W

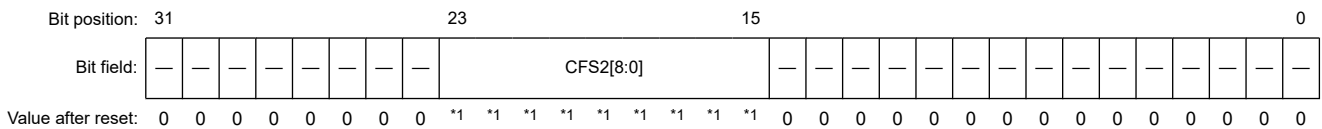
Note: S-TYPE-2, P-TYPE-1.

The MSPAR register specifies the privilege attribution for the corresponding bit in the Module Stop Control Register.

### 40.8.11 CFSAMONA : Code Flash Security Attribution Monitor Register A

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x30



Note 1. The value in a blank product is 0x1FF. It is set to the value written by your application.

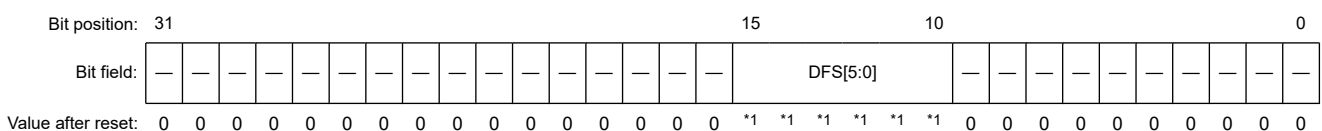
Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0.	R
23:15	CFS2[8:0]	Code Flash Secure Area Indicate the area of secure region for code flash.	R
31:24	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-5.

### 40.8.12 DFSAMON : Data Flash Security Attribution Monitor Register

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x34



Note 1. The value in a blank product is 0x3F. It is set to the value written by your application.

Bit	Symbol	Function	R/W
9:0	—	These bits are read as 0.	R
15:10	DFS[5:0]	Data Flash Secure Area Indicate the area of secure region for data flash.	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-5.

### 40.8.13 DLMMON : Device Lifecycle Management State Monitor Register

Base address: PSCU = 0x4020\_4000  
PSCU\_NS = 0x5020\_4000

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DLMMON[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1

Note 1. The value in a blank product is 0100b. These bits depend on DLM status.

Bit	Symbol	Function	R/W
3:0	DLMMON[3:0]	Device Lifecycle Management State Monitor Indicate DLM status value.  0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: OEM 0x5: Reserved 0x6: LCK_BOOT 0x7: RMA_REQ 0x8: RMA_ACK 0x9: RMA_RET Others: Reserved	R
31:4	—	These bits are read as 0.	R

Note: S-TYPE-5, P-TYPE-5.

### 40.8.14 MSAOAD : Master Security Attribution Operation After Detection Register

Base address: BUS = 0x4000\_3000

Offset address: 0x1010

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after detection 0: NMI 1: Reset	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writing of the OAD bit.	W

Note: S-TYPE-6, P-TYPE-2.

### OAD bit (Operation after detection)

The OAD bit specifies operation when the access violation is detected. When OAD = 0, error response is returned and NMI is generated. When OAD = 1, reset request is generated. When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits using halfword access.

### KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the OAD bit. When writing to the OAD bit, write 0xA5 simultaneously to the KEY[7:0] bits. If this value is not written to the KEY[7:0] bits, the OAD bit is not updated.

## 40.8.15 MSAPT : Master Security Attribution Protect Register

Base address: BUS = 0x4000\_3000

Offset address: 0x1014

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	PROTECT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PROTECT	Protection of register 0: MSAOAD register writing is possible. 1: MSAOAD register writing is protected. Read is possible.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits enable or disable writing of the PROTECT bit.	W

Note: S-TYPE-6, P-TYPE-2.

### PROTECT bit (Protection of register)

The PROTECT bit enables or disables writes to the associated registers to be protected.

MSAPT.PROTECT controls the MSAOAD register.

When the PROTECT bit is set simultaneously, write 0xA5 to the KEY[7:0] bits using half word access.

### KEY[7:0] bits (Key Code)

The KEY[7:0] bits enable or disable writing to the PROTECT bit. When writing to the PROTECT bit, write 0xA5 simultaneously to the KEY[7:0] bits. If this value is not written to the KEY[7:0] bits, the PROTECT bit is not updated.

## 40.9 Usage Note

### 40.9.1 Security or Privilege Bit Write Timing

When writing a security or privilege attribution bit, confirm that the write is complete by reading the security or privilege register until it matches the value written. The protection is not effective until the write to the security or privilege register is complete.

## 41. Renesas Secure IP (RSIP-E51A)

### 41.1 Overview

This security engine consists of an isolated subsystem comprising an access management circuit, storage area, encryption/decryption circuit, and random number generation circuit. In combination with the RSIP library, the security engine can prevent eavesdropping (to maintain confidentiality), falsification of information (to ensure integrity), and impersonation (to verify authenticity).

Because key information required for encryption and decryption is stored only in the security engine and all accesses from the outside can be blocked, the security engine enables building a more robust security system.

Only access control circuit, random number generation circuit, and unique ID are supported. The operation of other circuits is not guaranteed.

Table 41.1 lists the security engine specifications. Figure 41.1 shows the security engine block diagram.

**Table 41.1 Security engine specifications (1 of 2)**

Parameter	Specifications
Access control	Access management circuit <ul style="list-style-type: none"> <li>In case of irregular access to the security engine due to a tampered program or CPU runaway, this circuit blocks all subsequent accesses and stops data output from the security engine</li> </ul>
Symmetric cryptography	AES: Compliant with NIST FIPS PUB 197 <ul style="list-style-type: none"> <li>Key length: 128, 192, or 256 bits</li> <li>Data block size: 128 bits</li> <li>AES supports following block cipher mode ECB, CBC, CTR: Compliant with NIST SP 800-38A</li> <li>CCM: Compliant with NIST SP 800-38C</li> <li>GCM: Compliant with NIST SP 800-38D</li> <li>XTS: Compliant with IEEE 1619-2007</li> <li>AES supports following authentication algorithm CMAC: Compliant with NIST SP 800-38B</li> <li>GMAC: Compliant with NIST SP 800-38D</li> <li>AES does not support the KEY size of 192-bit in CCM, GCM, XTS, CMAC and GMAC.</li> </ul>
Random number generation	128-bit true random number generation circuit
Asymmetric cryptography	RSA <ul style="list-style-type: none"> <li>Maximum number of operable bits: 4224 bits</li> <li>Supported key sizes: 1024-bit, 2048-bit, 3072-bit, and 4096-bit</li> <li>Signature generation, signature verification, public-key encryption, private-key decryption</li> </ul> ECC <ul style="list-style-type: none"> <li>Maximum number of operable bits: 576 bits</li> <li>Support for curve               <ul style="list-style-type: none"> <li>NIST P-192, P-224, P-256, P-384 and P-521</li> <li>Brainpool P256r1, P384r1, and P512r1</li> <li>Ed25519</li> <li>secp256k1</li> </ul> </li> <li>Signature generation, signature verification, key generation</li> </ul>
Message digest computation	HASH <ul style="list-style-type: none"> <li>Block size:               <ul style="list-style-type: none"> <li>512-bit(SHA-224, SHA-256)</li> <li>1024-bit(SHA-512/224, SHA-512/256, SHA-384, SHA-512)</li> </ul> </li> <li>Key size: 512-bit or less</li> <li>HASH supports following secure hash algorithms SHA-224, SHA-256, SHA-512/224, SHA-512/256, SHA-384 and SHA-512: Compliant with FIPS PUB 180-4</li> <li>HASH supports following message authentication algorithm HMAC: Compliant with FIPS PUB 198</li> </ul>

**Table 41.1 Security engine specifications (2 of 2)**

Parameter	Specifications
Hardware Unique Key	<ul style="list-style-type: none"> <li>• A read-only, 256-bit Hardware Unique Key (HUK) is exclusively accessible by the security engine access management circuit through a dedicated bus.</li> <li>• Key derivation functions (KDFs) combine the Hardware Unique Key with the key generation information. The derived keys implement the key wrapping for user key secure storage.</li> <li>• The HUK uniqueness prevents the illicit cloning and copying of keys to another MCU of the MCU group.</li> <li>• The HUK itself is stored in wrapped (encrypted, non-plain) format in an isolated memory area. Therefore, it is protected from illicit access and copy.</li> </ul>
Application Key Management	<ul style="list-style-type: none"> <li>• Wrapped keys are only valid within the security engine.</li> </ul>
Unique ID	<ul style="list-style-type: none"> <li>• A read-only, 128-bit ID unique to an MCU (Unique ID) is accessible from the access management circuit.</li> <li>• Key derivation functions (KDFs) combine the Unique ID with the key generation information. Such derived keys are used to unwrap the HUK within the security engine boundary.</li> </ul>
Decryption-on-the-fly	The security engine outputs the key data of the decryption-on-the-fly IP (DOTF) through a dedicated bus
Tamper Resistance	Countermeasures available for side-channel attacks, including SPA/DPA and timing attacks.
Module-stop function	Module-stop state can be set to reduce power consumption
TrustZone Filter	Security and Privilege attribution can be set

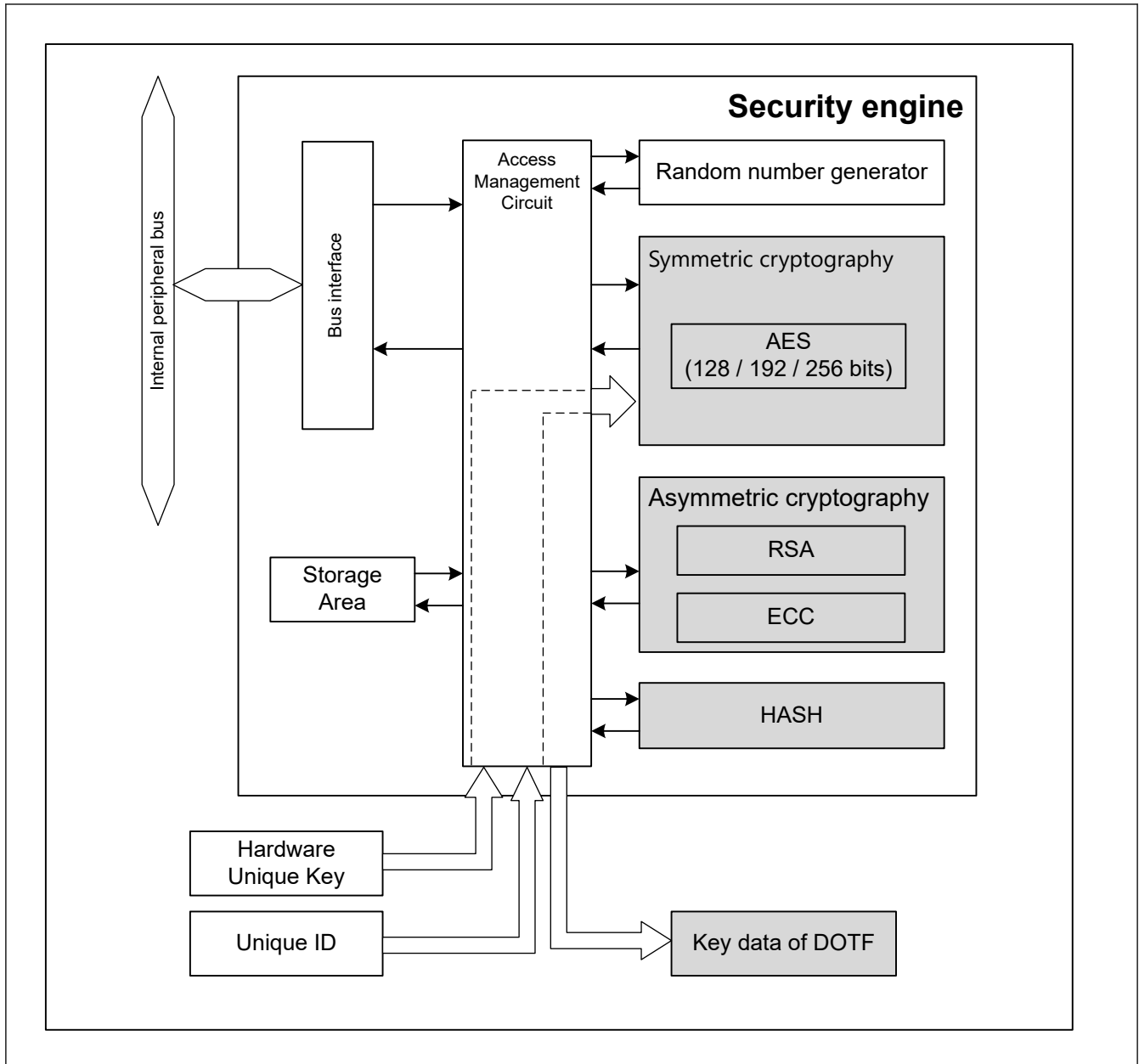


Figure 41.1 Security engine block diagram

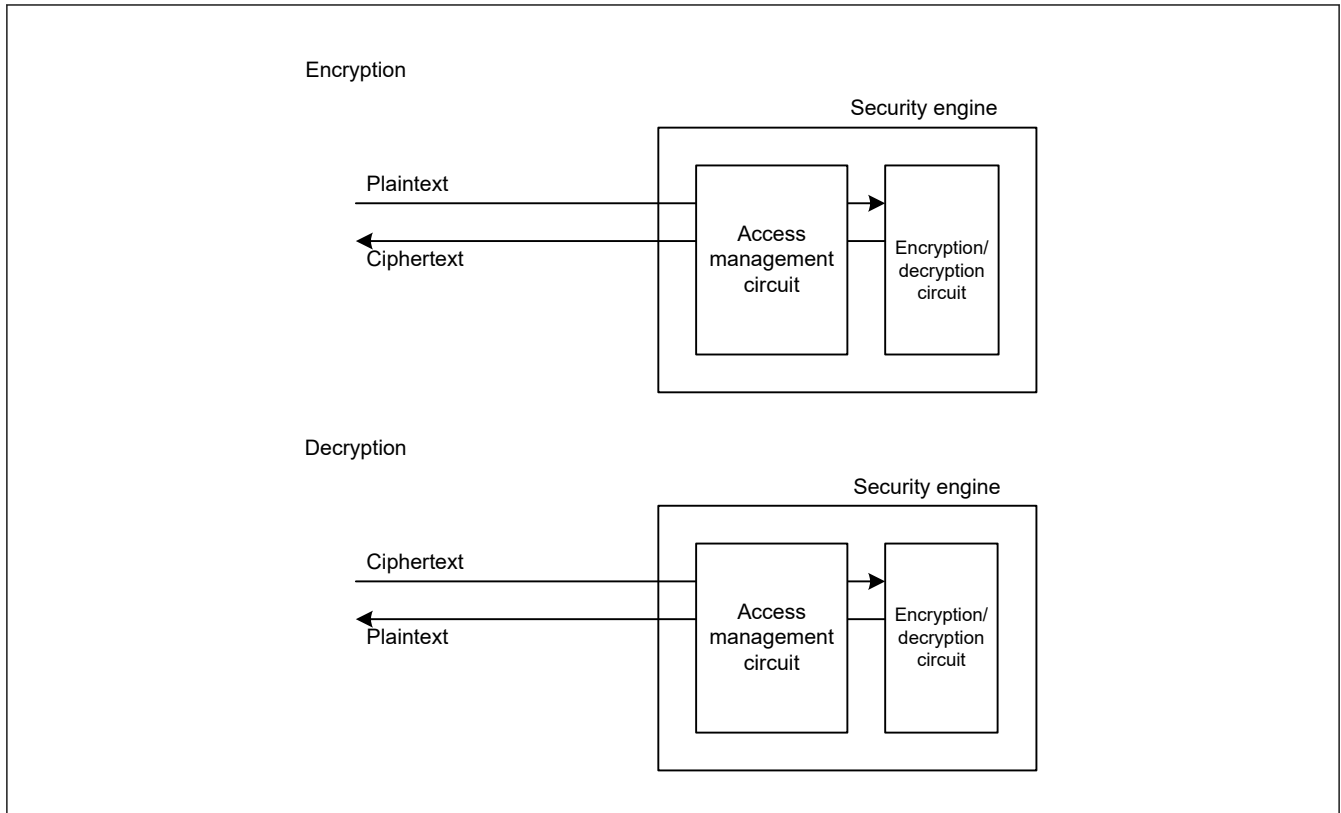
## 41.2 Operation

### 41.2.1 Symmetric cryptography

Figure 41.2 shows conceptual block diagram of the symmetric cryptography installed in the security engine.

The symmetric cryptography uses the input key information and converts the plaintext data to ciphertext or ciphertext data to plaintext through the hardware.

The encryption/decryption process can be completed without exposing the key data and the process's intermediate data outside of the security engine. This process is performed by the encryption/decryption circuit and storage area of the security engine internally.



**Figure 41.2 Conceptual diagram of the symmetric cryptography**

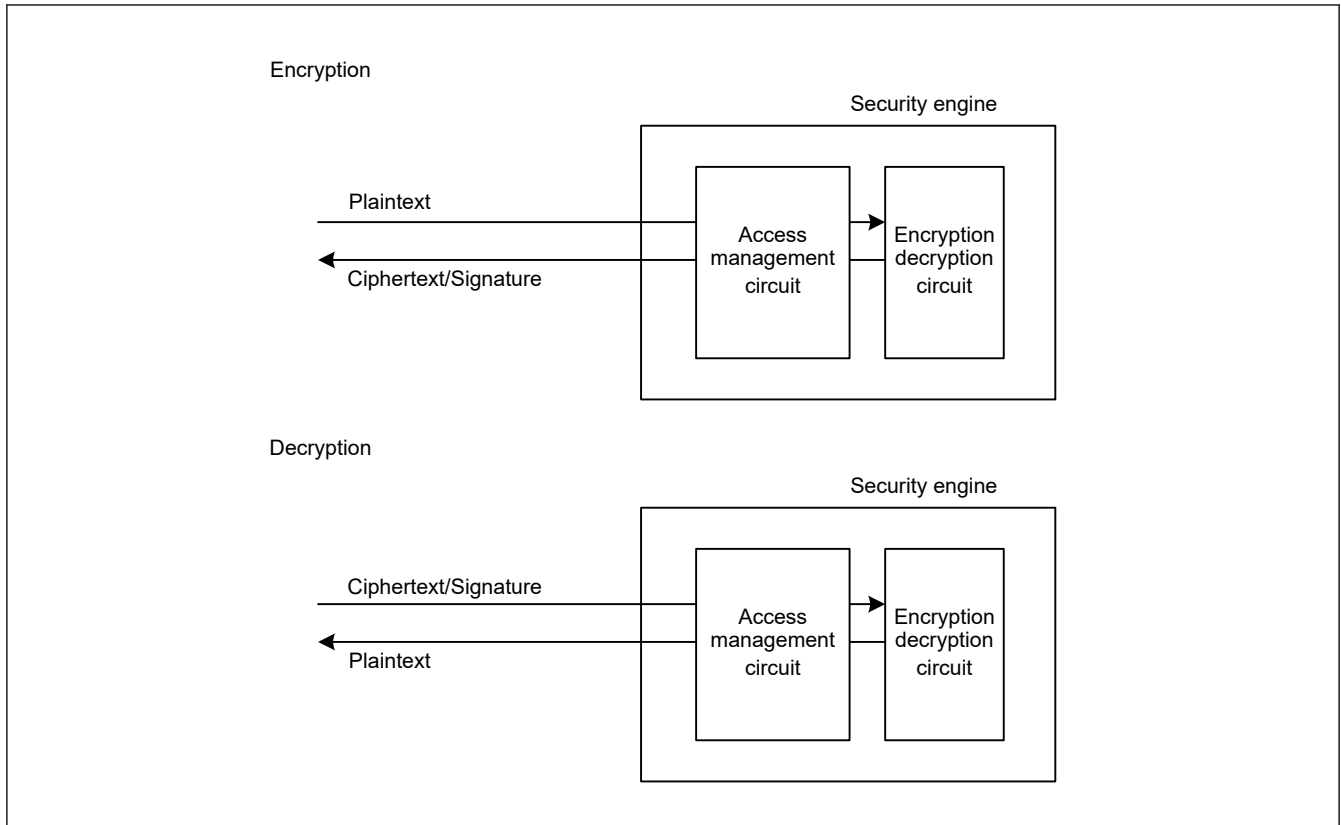
### 41.2.2 Asymmetric Cryptography

Figure 41.3 shows conceptual block diagram of the asymmetric cryptography installed in the security engine.

The asymmetric cryptography uses the input key information and converts the plaintext data to ciphertext or ciphertext data to plaintext or generates/verifies the signature.

The encryption/decryption or signature generation/verification process can be completed without exposing the key data and the process's intermediate data outside of the security engine. This process is performed by the cryptographic circuit and storage area of the security engine internally.





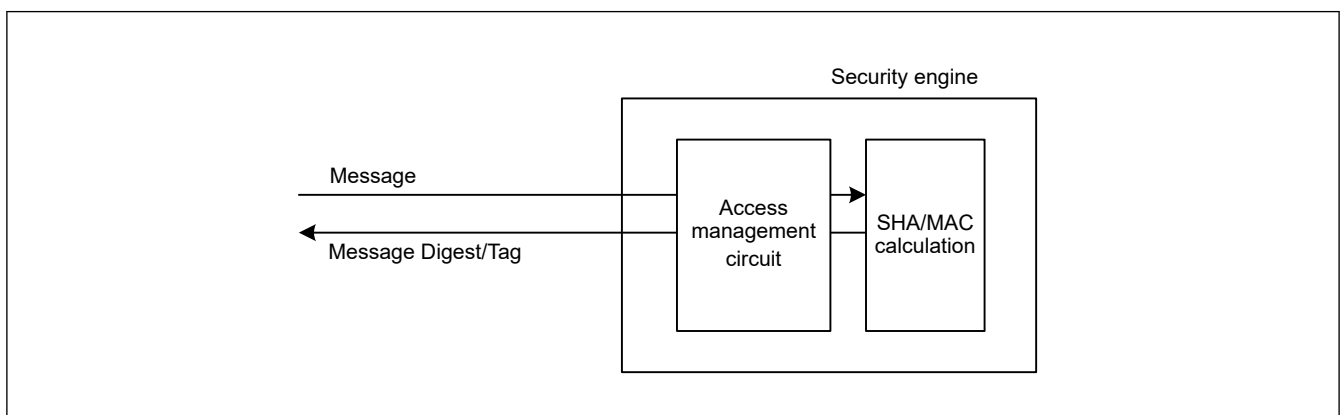
**Figure 41.3** Conceptual diagram of the asymmetric cryptography

### 41.2.3 Hash generator

Figure 41.4 shows conceptual block diagram of the hash generator installed in the security engine.

The hash generator generates uses the input key information (if required for the hash calculation) and generates the message digest or tag.

The calculation process can be completed without exposing any key data and the process's intermediate data outside of the security engine. This process is performed by the hash circuit and storage area of the security engine internally.



**Figure 41.4** Conceptual diagram of the hash generator

### 41.2.4 Encryption and Decryption

Follow the procedure below to encrypt and decrypt the data:

1. Input the key information to the security engine.
2. Input the target data to the security engine. Plaintext data is converted to ciphertext and ciphertext data to plaintext.
3. Read the converted data.

The encryption/decryption circuit has input and output buffers, and can perform encryption/decryption in parallel with data input/output. Figure 41.5 shows the encryption engine timing.

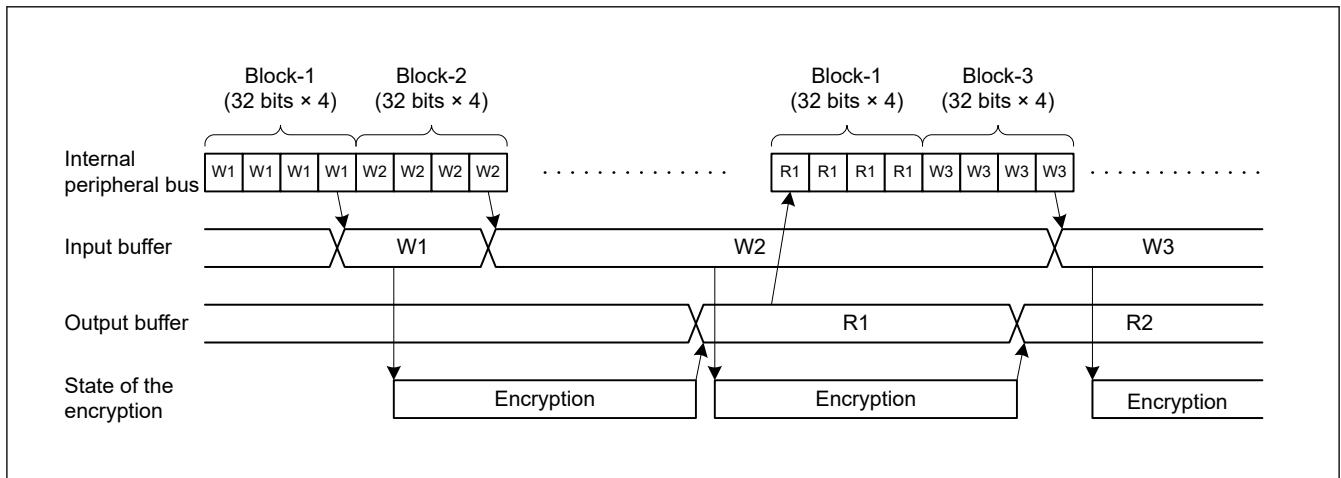


Figure 41.5 Encryption and decryption timing (AES)

### 41.3 Usage Notes

#### 41.3.1 Software Standby Mode

When Software Standby mode is entered while the security engine is in process, proper processing cannot be resumed after cancelling Software Standby mode. Software Standby mode should therefore be entered while the security engine is not running.

#### 41.3.2 Module-Stop Function Setting

The security engine operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The security engine module is initially stopped after reset. Releasing the module-stop state enables access to the registers.

#### 41.3.3 SPA/DPA Protections

SPA/DPA protections can optionally be enabled for encryption/decryption processing.

#### 41.3.4 Restrictions on Open process

While open process of security engine's API (R\_RSIP\_Open) is running, the following restrictions is applied.

- Available CPU frequency is up to 240MHz
- Do not change frequency of CPUCLK, ICLK and PCLKn (n = A to E)
- Do not change MSTPCRn registers (n = A to E)
- Do not transition to CPU Sleep mode, CPU Deep Sleep mode, Software Standby mode and Deep Software Standby mode 1/2/3."

## 42. 12-Bit A/D Converter (ADC12)

### 42.1 Overview

The MCU includes 12-bit successive approximation A/D converter (ADC12) units. In unit 0, up to 8 analog input channels are selectable. In unit 1, up to 5 analog input channels, temperature sensor output, internal reference voltage, VBATT 1/3 voltage monitor output can be selected for conversion in respective units.

The A/D conversion accuracy is selectable from 12-bit, 10-bit, 8-bit conversion, making it possible to optimize the trade-off between speed and resolution in generating a digital value.

The ADC12 supports the following operating modes:

- Single scan mode to convert analog inputs of selected channels in ascending order of channel number
- Continuous scan mode to convert analog inputs of selected channels continuously in ascending order of channel number
- Group scan mode to divide analog inputs of channels into two groups (group A and B) and convert the analog inputs of selected channels for each group in ascending order of channel number.

In group scan mode, select two groups (group A and B). You can individually select the scan start conditions for each group (group A, B) and start scanning of each group at different times. In addition, when group A priority control operation is set, the ADC12 accepts group A scan start during group B A/D conversion, suspending group B conversion. This allows you to assign higher priority to A/D conversion start for group A.

In double trigger mode, the analog input of a selected channel is converted in single scan mode or group scan mode (group A), and data converted by the first and second A/D conversion start triggers are stored in different registers, providing duplexing of A/D converted data.

Self-diagnosis is performed once at the beginning of each scan, and one of the three reference voltage values generated in ADC12 is A/D converted.

The temperature sensor output and the internal reference voltage is selectable at the same time as the analog input of the channel. First A/D conversion is performed for the analog input of the channel, next the temperature sensor output and then for the internal reference voltage.

The ADC12 also provides a compare function (window A and window B). The compare function specifies the upper reference value for window A and lower reference value for window B, and outputs an interrupt when the A/D converted value of the selected channel meets the comparison conditions.

The A/D data storage buffer is a ring buffer consisting of 16 buffers to sequentially store A/D converted data.

[Table 42.1](#) lists the ADC12 specifications and [Table 42.2](#) lists the functions. [Figure 42.1](#) shows a block diagram of ADC12 and [Table 42.3](#) lists the I/O pins.

**Table 42.1 ADC12 specifications (1 of 3)**

Parameter	Specifications
Number of units	Two units
Input channels	Up to 13 channels (AN000 to AN002, AN004 to AN008, AN100 to AN102, AN104, AN105)
Analog function	Temperature sensor output, internal reference voltage, VBATT 1/3 voltage monitor output
Conversion method	Successive approximation method
Resolution	12-bit, 10-bit, 8-bit
Conversion time	0.4 $\mu$ s/channel (when 12-bit A/D conversion clock PCLKC (ADCLK) is operating at 60 MHz)
A/D conversion clock	Peripheral module clock PCLKA and A/D conversion clock PCLKC (ADCLK) can be set with the following division ratios: PCLKA to PCLKC (ADCLK) frequency ratio = 1:1, 2:1, 4:1, 8:1, 1:2, 1:4

Table 42.1 ADC12 specifications (2 of 3)

Parameter	Specifications
Data registers <sup>*1</sup>	<ul style="list-style-type: none"> <li>• 13 registers for analog input</li> <li>• One register for A/D-converted data duplication in double trigger mode</li> <li>• Two registers for A/D-converted data duplication during extended operation in double trigger mode</li> <li>• One register for temperature sensor output</li> <li>• One register for internal reference voltage</li> <li>• One register for VBATT 1/3 voltage monitor output</li> <li>• One register for self-diagnosis</li> <li>• A/D conversion results are stored in A/D data registers</li> <li>• 12-bit, 10-bit, 8-bit accuracy for A/D conversion results</li> <li>• A/D-converted value addition mode, in which the sum of all A/D-converted results is stored in the A/D data registers as a value with the conversion accuracy bit count + extended bits</li> <li>• Double-trigger mode (selectable in single scan and group scan modes): <ul style="list-style-type: none"> <li>– The first unit of A/D-converted analog input data on one selected channel is stored in the data register for the channel, and the second unit is stored in the duplication register.</li> </ul> </li> <li>• Extended operation in double trigger mode (available for specific triggers): <ul style="list-style-type: none"> <li>– A/D-converted analog input data on one selected channel is stored in the duplication register provided for the associated trigger.</li> </ul> </li> </ul>
Operating modes <sup>*2</sup>	<ul style="list-style-type: none"> <li>• Single scan mode: <ul style="list-style-type: none"> <li>– A/D conversion is performed only once on the analog inputs of arbitrarily selected channels, on the temperature sensor output, on the internal reference voltage.</li> </ul> </li> <li>• Continuous scan mode: <ul style="list-style-type: none"> <li>– A/D conversion is performed repeatedly on the analog inputs of the selected channels on the temperature sensor output, and on the internal reference voltage.</li> </ul> </li> <li>• Group scan mode: <ul style="list-style-type: none"> <li>– Analog inputs of selected channels, the temperature sensor output, and the internal reference voltage are divided into groups A and B. Then A/D conversion of the analog inputs selected on a group basis is performed once.</li> <li>– The scan start conditions can be independently selected for group A, B, allowing A/D conversion of group A, B to be started independently.</li> </ul> </li> <li>• Group scan mode (when group priority operation is selected): <ul style="list-style-type: none"> <li>– If a priority group trigger is input during scanning of a lower-priority group, the scanning of the lower-priority group is stopped and then scanning of the priority group is started. The order of priority is group A &gt; group B.</li> <li>– It is possible to select whether to restart scanning (rescan) of the lower-priority group upon completion of the priority group scan. It is also possible to specify rescanning to be started from the first channel of the selected channels or from the channel for which A/D conversion has not been completed.</li> </ul> </li> </ul>
Conditions for A/D conversion start	<ul style="list-style-type: none"> <li>• Software trigger</li> <li>• Synchronous triggers from the Event Link Controller (ELC) and GPT</li> <li>• Asynchronous triggering by the external trigger pins, ADTRG0 (unit 0) and ADTRG1 (unit 1)</li> </ul>
Functions	<ul style="list-style-type: none"> <li>• Dedicated sample-and-hold function with optional constant sampling and 2 channels in unit 0</li> <li>• Variable sampling state count</li> <li>• Self-diagnosis of A/D converter</li> <li>• Selectable A/D-converted value addition mode or average mode</li> <li>• Analog input disconnection detection function (discharge and precharge functions)</li> <li>• Double-trigger mode (duplication of A/D conversion data)</li> <li>• Automatic clear function for A/D data registers</li> <li>• Digital comparison of values in the comparison register and data register, and comparison between values in the data registers</li> <li>• Ring buffer</li> </ul>

**Table 42.1 ADC12 specifications (3 of 3)**

Parameter	Specifications
Interrupt sources	<ul style="list-style-type: none"> <li>In single scan mode (double trigger deselected), an A/D scan end interrupt request (ADC12i_ADI (i = 0, 1)) and ELC event signal (ADC12i_ADI (i = 0, 1)) can be generated on completion of single scan. <ul style="list-style-type: none"> <li>A compare interrupt request (ADC12i_CMPAI (i = 0, 1)/ADC12i_CMPBI (i = 0, 1)) can be generated in response to a match with a digital comparison condition.</li> <li>A window compare ELC event signal (ADC12i_WCMPI (i = 0, 1)) can be generated in response to a match with a digital comparison condition.</li> <li>A window compare ELC event signal (ADC12i_WCMPUM (i = 0, 1)) can be generated in response to a mismatch with a digital comparison condition.</li> </ul> </li> <li>In single scan mode (double trigger selected), an A/D scan end interrupt request (ADC12i_ADI (i = 0, 1)) and ELC event signal (ADC12i_ADI (i = 0, 1)) is generated on completion of two scans.</li> <li>In continuous scan mode, an A/D scan end interrupt request (ADC12i_ADI (i = 0, 1)) and ELC event signal (ADC12i_ADI (i = 0, 1)) is generated on completion of all the selected channel scans.</li> <li>In group scan mode (double trigger deselected), an A/D scan end interrupt request (ADC12i_ADI (i = 0, 1)) and ELC event signal (ADC12i_ADI (i = 0, 1)) is generated on completion of group A scan, and an A/D scan end interrupt request for group B (ADC12i_GBADI (i = 0, 1)) can be generated on completion of group B scan.</li> <li>In group scan mode (double trigger selected), an A/D scan end interrupt request (ADC12i_ADI (i = 0, 1)) and ELC event signal (ADC12i_ADI (i = 0, 1)) is generated on completion of two group A scans, and an A/D scan end interrupt request for group B (ADC12i_GBADI (i = 0, 1)) can be generated on completion of group B scan.</li> <li>ADC12i_ADI (i = 0, 1), ADC12i_GBADI (i = 0, 1), ADC12i_WCMPI (i = 0, 1), and ADC12i_WCMPUM (i = 0, 1) can activate the Data Transfer Controller (DTC).</li> </ul>
ELC interface	<ul style="list-style-type: none"> <li>An event is generated upon completion of group A scan in group-scan mode.</li> <li>An event is generated upon completion of group B scan in group-scan mode.</li> <li>An event is generated when all scans complete.</li> <li>Scan can be started by a trigger from the ELC.</li> <li>An event is generated according to conditions of the compare function window in single-scan mode.</li> </ul>
Reference voltage	<ul style="list-style-type: none"> <li>Unit 0: VREFH0 is the analog reference voltage. VREFL0 is the analog reference ground.</li> <li>Unit 1: VREFH1 is the analog reference voltage. VREFL1 is the analog reference ground.</li> </ul>
Analog channel input SA judgement function	Input channels of analog can be valid only when the security attribution of ADC12n (n = 0, 1) match the security attribution of PORT (PmSAR). When the security attribution of ADC12n (n = 0, 1) does not match the PORT security attribution, the conversion result of ADC input channel will be unexpected. Internal reference voltage, Temperature sensor output and VBATT 1/3 voltage monitor output have no Security Attribution judgement function.
Module-stop function	Module-stop state can be set to for each units reduce power consumption.*3
TrustZone Filter	Security and Privilege attribution can be set for each unit

Note 1. Changing the A/D conversion accuracy also changes the A/D conversion time. For details, see [section 42.3.6. Analog Input Sampling and Scan Conversion Time](#).

Note 2. When selecting the temperature sensor output, the internal reference voltage, do not use continuous scan mode or group scan mode.

Note 3. For details, see [section 10, Low Power Modes](#).

**Table 42.2 ADC12 functions (1 of 2)**

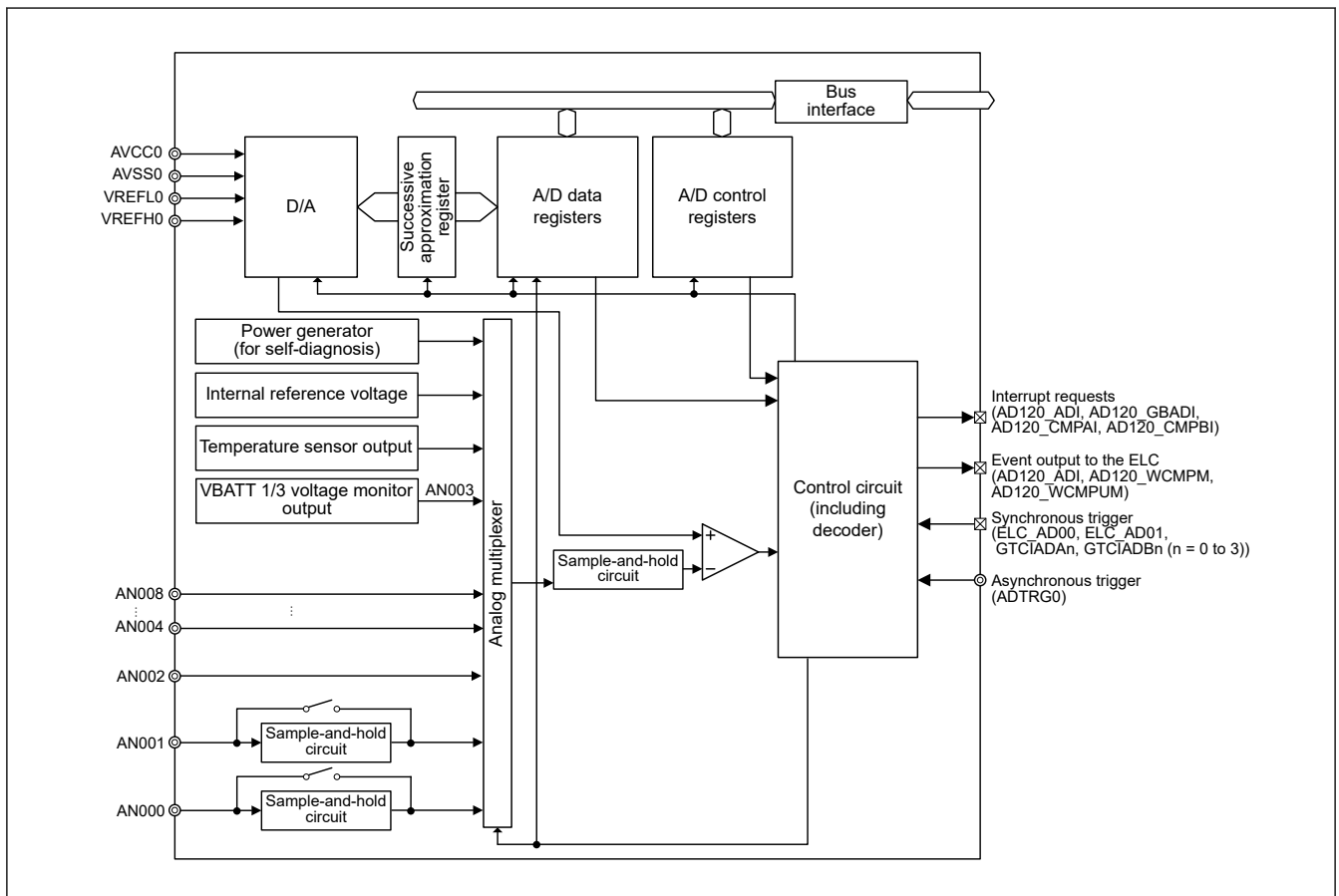
Parameter	Function
Analog input channel	AN000 to AN002, AN004 to AN008 (unit 0), AN100 to AN102, AN104, AN105 (unit 1) Internal reference voltage Temperature sensor output VBATT 1/3 voltage monitor output

**Table 42.2 ADC12 functions (2 of 2)**

Parameter			Function
Conditions for A/D conversion starts	Software	Software trigger	Enabled
	Asynchronous trigger (external trigger)	Trigger input pin	ADTRGn (n = 0, 1)
	Synchronous trigger (trigger from ELC)	ELC trigger	ELC_AD00 (unit 0) and ELC_AD10 (unit 1), ELC_AD01 (unit 0) and ELC_AD11 (unit 1)
GPT trigger		GTCIADAm, GTCIADBm (unit 0) (m = 0 to 3) GTCIADAn, GTCIADBn (unit 1) (n = 4, 5)	
Channel-dedicated sample-and-hold function Target channel			AN000, AN001 (unit 0)
Interrupt			ADC12i_ADI (i = 0, 1) ADC12i_GBADI (i = 0, 1) ADC12i_CMPAI (i = 0, 1) ADC12i_CMPBI (i = 0, 1)
Output to ELC			ADC12i_ADI (i = 0, 1) ADC12i_WCMPI (i = 0, 1) ADC12i_WCMPUM (i = 0, 1)
Module-stop function settings *1 *2			MSTPCRD.MSTPD16 bit (unit 0) MSTPCRD.MSTPD15 bit (unit 1)

Note 1. For details, see [section 10, Low Power Modes](#).

Note 2. Wait 1 μs or longer to start A/D conversion after release from the module-stop state.



**Figure 42.1 ADC12 block diagram (unit 0)**

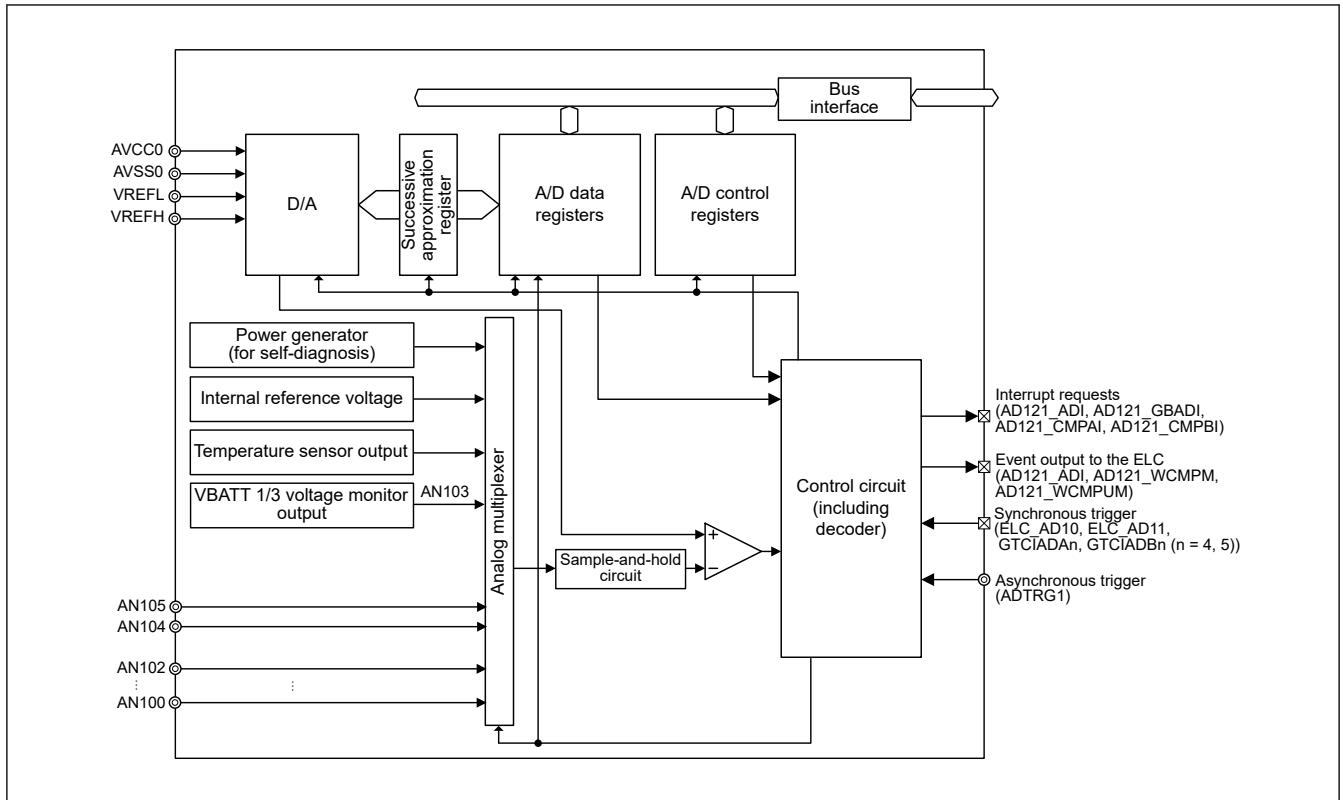


Figure 42.2 ADC12 block diagram (unit 1)

Table 42.3 lists the ADC12 I/O pins.

Table 42.3 ADC12 I/O pins (unit 0)

Pin name	I/O	Function
AVCC0	Input	Analog block power supply pin (Connect to VCC when ADC12/DAC12 is not used.)
AVSS0	Input	Analog block power supply ground pin (Connect to VSS when ADC12/DAC12 is not used.)
VREFH0	Input	Analog reference voltage supply pin
VREFL0	Input	Analog reference ground pin
AN000 to AN002, AN004 to AN008	Input	Analog input pins 0 to 2, 4 to 8
ADTRG0	Input	External trigger input pin for starting A/D conversion

Table 42.4 ADC12 I/O pins (unit 1)

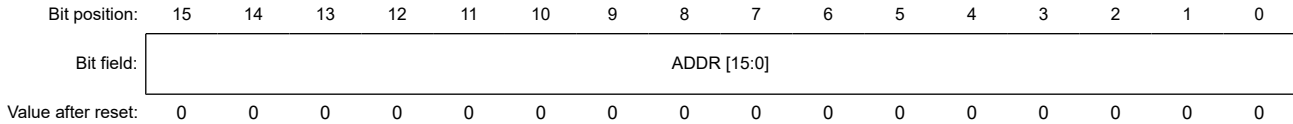
Pin name	I/O	Function
AVCC0	Input	Analog block power supply pin
AVSS0	Input	Analog block power supply ground pin
VREFH	Input	Analog reference voltage supply pin
VREFL	Input	Analog reference ground pin
AN100 to AN102, AN104, AN105	Input	Analog input pins 0 to 2, 4, 5
ADTRG1	Input	External trigger input pin for starting A/D conversion

## 42.2 Register Descriptions

### 42.2.1 ADDRn : A/D Data Registers n

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address:  $0x020 + 0x2 \times n$  ( $n = 0$  to  $8, m = 0$ )  
 $0x020 + 0x2 \times n$  ( $n = 0$  to  $5, m = 1$ )



Bit	Symbol	Function	R/W
15:0	ADDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See <a href="#">Table 42.5</a> and <a href="#">Table 42.6</a> .	R

Note: S-TYPE-3, P-TYPE-3

ADDRn registers are 16-bit read-only registers to store A/D conversion results.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (12-bit, 10-bit, 8-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

#### (1) When A/D-converted value addition/average mode is not selected

[Table 42.5](#) shows the example of bit assignment for 12-bit accuracy.

**Table 42.5 Example of bit assignment for 12-bit accuracy**

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

#### (2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, these registers indicate the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

#### (3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, these registers indicate the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.



Table 42.6 shows example of the bit assignment for 12-bit accuracy.

**Table 42.6 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected**

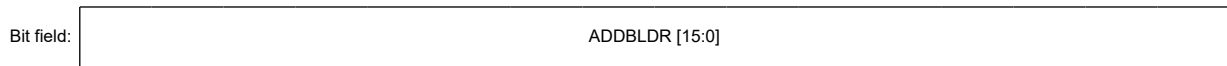
Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

### 42.2.2 ADDBLDR : A/D Data Duplexing Register

Base address: ADC12m = 0x4033\_2000 + 0x0200 × m (m = 0, 1)  
 ADC12m\_NS = 0x5033\_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x018

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	ADDBLDR [15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 42.7 and Table 42.8.	R

Note: S-TYPE-3, P-TYPE-3

ADDBLDR register is a 16-bit read-only register to store A/D conversion results in response to a second trigger in double-trigger mode.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (12-bit, 10-bit, 8-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

#### (1) When A/D-converted value addition/average mode is not selected

Table 42.7 shows the example of bit assignment for 12-bit accuracy.

**Table 42.7 Example of bit assignment for 12-bit accuracy**

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.			Converted Value 11 to 0: 12-bit A/D-converted value												
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, this register indicates the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 42.8 shows example of the bit assignment for 12-bit accuracy.

**Table 42.8 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected**

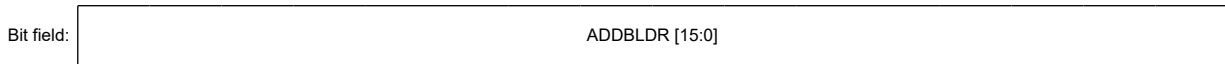
Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.				Added Value 13 to 0: 14-bit sum of A/D conversion results											
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

42.2.3 ADDBLDRn : A/D Data Duplexing Register n (n = A, B)

Base address: ADC12m = 0x4033\_2000 + 0x0200 × m (m = 0, 1)  
 ADC12m\_NS = 0x5033\_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x084 (n = A)  
 0x086 (n = B)

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	ADDBLDR [15:0]	Converted Value 15 to 0  Functions vary depending on the selected mode and accuracy. See Table 42.9 and Table 42.10.	R

Note: S-TYPE-3, P-TYPE-3

ADDBLDRn registers are 16-bit read-only registers to store A/D conversion results in response to respective triggers during extended operation in double-trigger mode.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (12-bit, 10-bit, 8-bit is selectable.)

- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 42.9 shows the example of bit assignment for 12-bit accuracy.

**Table 42.9 Example of bit assignment for 12-bit accuracy**

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, these registers indicate the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, these registers indicate the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 42.10 shows example of the bit assignment for 12-bit accuracy.

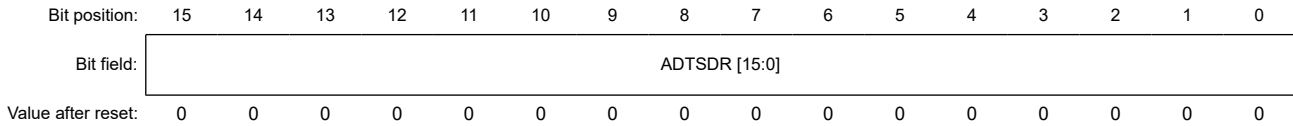
**Table 42.10 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected**

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified		Added Value 15 to 0: 16-bit sum of A/D conversion results													
	When 1, 2, 3, or 4 conversion times is specified		These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results											
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified		Added Value 15 to 0: 16-bit sum of A/D conversion results													
	When 16 conversion times is specified		Added Value 13 to 0: 14-bit sum of A/D conversion results													These bits are read as 0.

### 42.2.4 ADTSDR : A/D Temperature Sensor Data Register

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x01A



Bit	Symbol	Function	R/W
15:0	ADTSDR [15:0]	Converted Value 15 to 0  Functions vary depending on the selected mode and accuracy. See <a href="#">Table 42.11</a> and <a href="#">Table 42.12</a> .	R

Note: S-TYPE-3, P-TYPE-3

ADTSDR register is a 16-bit read-only register to store A/D conversion result of the temperature sensor output.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (12-bit, 10-bit, 8-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

#### (1) When A/D-converted value addition/average mode is not selected

[Table 42.11](#) shows the example of bit assignment for 12-bit accuracy.

**Table 42.11 Example of bit assignment for 12-bit accuracy**

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

#### (2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, this register indicates the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

#### (3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

[Table 42.12](#) shows example of the bit assignment for 12-bit accuracy.

**Table 42.12 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected**

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results													
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

### 42.2.5 ADOCDR : A/D Internal Reference Voltage Data Register

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x01C

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field: ADOCDR [15:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	ADOCDR [15:0]	Converted Value 15 to 0  Functions vary depending on the selected mode and accuracy. See <a href="#">Table 42.13</a> and <a href="#">Table 42.14</a> .	R

Note: S-TYPE-3, P-TYPE-3

ADOCDR register is a 16-bit read-only register to store A/D conversion result of the internal reference voltage.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (12-bit, 10-bit, 8-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

#### (1) When A/D-converted value addition/average mode is not selected

[Table 42.13](#) shows the example of bit assignment for 12-bit accuracy.

**Table 42.13 Example of bit assignment for 12-bit accuracy**

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.		Converted Value 11 to 0: 12-bit A/D-converted value														
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value														These bits are read as 0.		

#### (2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, this register indicates the mean of A/D-converted values on a

specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 42.14 shows example of the bit assignment for 12-bit accuracy.

**Table 42.14 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected**

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.				Added Value 13 to 0: 14-bit sum of A/D conversion results											
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

42.2.6 ADVMDR : A/D VBATT Monitor Data Register

Base address: ADC12m = 0x4033\_2000 + 0x0200 × m (m = 0, 1)  
 ADC12m\_NS = 0x5033\_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x026

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field: ADDR [15:0]

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
15:0	ADDR[15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See Table 42.15 and Table 42.16	R

Note: S-TYPE-3, P-TYPE-3

ADVMDR register is a 16-bit read-only register to store A/D conversion result of the VBATT 1/3 voltage. For VBATT 1/3 voltage measurement, VBATTMNSLR.VBTMNSSEL need to be set. For more information, see section 11, Battery Backup Function. ADVMDR register is an alias for the ADDR3 register.

The following conditions determine the formats for data in the A/D data registers:

- Setting of the A/D Data Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- The setting in the A/D Conversion Accuracy Select bits (ADCER.ADPRC[1:0]) (12-bit, 10-bit, 8-bit is selectable.)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

(1) When A/D-converted value addition/average mode is not selected

Table 42.15 shows the example of bit assignment for 12-bit accuracy.

**Table 42.15 Example of bit assignment for 12-bit accuracy**

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

(2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, this register indicates the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

(3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, this register indicates the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 42.16 shows example of the bit assignment for 12-bit accuracy.

**Table 42.16 Example of bit assignment for 12-bit accuracy when A/D-converted value addition mode is selected**

Accuracy		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 1, 2, 3, or 4 conversion times is specified	These bits are read as 0.			Added Value 13 to 0: 14-bit sum of A/D conversion results												
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified	Added Value 15 to 0: 16-bit sum of A/D conversion results															
	When 16 conversion times is specified	Added Value 13 to 0: 14-bit sum of A/D conversion results														These bits are read as 0.	

42.2.7 ADDR : A/D Self-Diagnosis Data Register

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x01E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DIAGST[1:0]		—	—	AD[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	AD[11:0]	Converted Value 11 to 0 12-bit A/D-converted value	R
13:12	—	These bits are read as 0.	R
15:14	DIAGST[1:0]	Self-Diagnosis Status For details on self-diagnosis, see <a href="#">section 42.2.13. ADCER : A/D Control Extended Register</a> . 0 0: Self-diagnosis not executed after power-on. 0 1: Self-diagnosis was executed using the 0 V voltage. 1 0: Self-diagnosis was executed using the reference voltage <sup>*1</sup> × 1/2. 1 1: Self-diagnosis was executed using the reference voltage <sup>*1</sup> .	R

Note: S-TYPE-3, P-TYPE-3

Note: The example of the bit assignment for the right-justified data with 12-bit accuracy is indicated.

Note 1. The reference voltage refers to VREFH0 for unit 0 and to VREFH for unit 1.

ADRD is a 16-bit read-only register that holds the A/D conversion results based on the self-diagnosis of the ADC12. In addition to the AD[11:0] bits indicating the A/D-converted value, it includes the Self-Diagnosis Status bit (DIAGST[1:0]).

The settings of the A/D data register format and the A/D conversion accuracy determines the formats for data in this register.

The A/D-converted value addition and average modes cannot be applied to the A/D self-diagnosis function. For details on self-diagnosis, see [section 42.2.13. ADCER : A/D Control Extended Register](#).

This section describes the data formats for each condition. The register diagram and the register bit table shown in this section indicate example of the bit assignment for the left and right-justified data with 12-bit accuracy.

**Table 42.17 Bit assignment for each right-justified accuracy**

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	DIAGST[1:0]	—	—	—	AD[11:0]	—	—	—	—	—	—	—	—	—	—	—

**Table 42.18 Bit assignment for each left-justified accuracy**

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Left-justified data with 12-bit accuracy	AD[11:0]	—	—	—	—	—	—	—	—	—	—	—	—	—	DIAGST[1:0]	—

### 42.2.8 ADCSR : A/D Control Register

Base address: ADC12m = 0x4033\_2000 + 0x0200 × m (m = 0, 1)  
ADC12m\_NS = 0x5033\_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x000

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	ADST	ADCS[1:0]	—	—	—	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]				
------------	------	-----------	---	---	---	------	-------	------	--------	---	-------------	--	--	--	--

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
4:0	DBLANS[4:0]	Double Trigger Channel Select These bits select one analog input channel for double-trigger operation. The setting is only valid in double-trigger mode.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	GBADIE	Group B Scan End Interrupt and ELC Event Enable Group B scan only works in group scan mode. 0: Disable ADC12i_GBADI (i = 0, 1) interrupt generation on group B scan completion. 1: Enable ADC12i_GBADI (i = 0, 1) interrupt generation on group B scan completion.	R/W



Bit	Symbol	Function	R/W
7	DBLE	Double Trigger Mode Select 0: Deselect double-trigger mode. 1: Select double-trigger mode.	R/W
8	EXTRG	Trigger Select*1 0: Start A/D conversion by the synchronous trigger (GPT, ELC). 1: Start A/D conversion by the asynchronous trigger (ADTRGn*2).	R/W
9	TRGE	Trigger Start Enable 0: Disable A/D conversion to be started by the synchronous or asynchronous trigger 1: Enable A/D conversion to be started by the synchronous or asynchronous trigger	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
14:13	ADCS[1:0]	Scan Mode Select 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
15	ADST	A/D Conversion Start 0: Stop A/D conversion process. 1: Start A/D conversion process.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. To start A/D conversion using an external pin (asynchronous trigger):

After a high-level signal is input to the external pin (ADTRGn), write 1 to both the TRGE and EXTRG bits in the ADCSR register and drive the ADTRGn pin low. With these settings, the scan conversion process starts on detection of the falling edge of ADTRGn. The pulse width of the low-level input must be at least PCLKA 1.5 clock cycles.

Note 2. n = 0 (unit 0), 1 (unit 1)

The ADCSR register sets double-trigger mode and A/D conversion start trigger, enables or disables scan end interrupt, selects the scan mode, and starts or stops A/D conversion.

#### DBLANS[4:0] bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one channel for A/D conversion data duplication in double-trigger mode. This can be selected by setting the binary value of the channel number to be duplicated. The A/D conversion results of the analog input of the channel selected in the DBLANS[4:0] bits are stored in A/D Data Register y when conversion is started by the first trigger, and stored in the A/D Data Duplexing Register when conversion is started by the second trigger.

In double-trigger mode, the channels selected in the ADANSA0 register, are invalid, and the channel selected in the DBLANS[4:0] bits is A/D converted instead.

When double-trigger mode is used in group scan mode, double-trigger control is only applied to group A and not to group B.

Also, in double trigger mode, the analog inputs of multiple channels, temperature sensor outputs, or internal reference voltage cannot be selected for group A, but can be selected for groups B.

Only set the DBLANS[4:0] bits when the ADST bit is 0. Do not set the DBLANS[4:0] bits at the same time that you write 1 to the ADST bit.

To enter A/D-converted value addition/average mode when in double-trigger mode, set the same channel in the ADADS0 register as the channel selected in the DBLANS[4:0] bits.

A/D-converted data from the self-diagnosis function temperature sensor output and internal reference voltage cannot be used in double-trigger mode.

#### GBADIE bit (Group B Scan End Interrupt and ELC Event Enable)

The GBADIE bit enables or disables group B scan end interrupt (ADC12i\_GBADI (i = 0, 1)) in group scan mode.

#### DBLE bit (Double Trigger Mode Select)

The DBLE bit selects or deselects double-trigger mode. Double-trigger mode can only be operated by the synchronous trigger (GPT, ELC) selected in the ADSTRGR.TRSA[5:0] bits.

Double-trigger operation is as follows:

1. The ADC12i\_ADI (i = 0, 1) interrupt is not output on completion of the first conversion but on completion of the second conversion.
2. The A/D conversion results from the duplication channel (selected in DBLANS[4:0]) started by the first trigger are stored in A/D Data Register y and those started by the second trigger are stored in the A/D Data Duplexing Register.

When the DBLE bit is set (double-trigger mode is selected), the channels specified in the ADANSA0 register are invalid. Double-trigger mode is deselected by setting DBLE to 0. Setting DBLE to 1 again enables the same double-trigger operation described in 1. and 2. for first time scanning with the first trigger.

Do not select double-trigger mode in continuous scan mode. When using double trigger mode in group scan mode, A/D conversion of the temperature sensor output and internal reference voltage should not be selected for group A. Software triggering cannot be used in double-trigger mode. Always set the ADST bit to 0 before setting the DBLE bit. Do not set the DBLE bit at that same time as writing 1 to the ADST bit.

### EXTRG bit (Trigger Select)

The EXTRG bit selects the synchronous or asynchronous trigger as the trigger for starting A/D conversion.

In group scan mode, the setting of this bit takes effect on the trigger selected for group A. For group B, A/D conversion is started by the selected synchronous trigger regardless of this bit setting.

### TRGE bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous and asynchronous triggers. In group scan mode, set this bit to 1.

### ADCS[1:0] bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.

In single scan mode, A/D conversion is performed for the analog inputs of the channels selected in the ADANSA0 register, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, the scan conversion stops.

In continuous scan mode, when the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs of the channels selected with the ADANSA0 register, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion repeats from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion stops even if scanning is in progress.

In group scan mode:

- Group A scanning is started by the synchronous trigger (GPT, ELC) selected in the TRSA[5:0] bits in the ADSTRGR register. A/D conversion is performed on group A analog inputs of the channels selected in the ADANSA0 register, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion stops.
- Group B scanning is started by the synchronous trigger (GPT, ELC) selected in the ADSTRGR.TRSB[5:0] bits. A/D conversion is performed on group B analog inputs of the channels selected in the ADANSB0 register, in ascending order of channel number. When 1 cycle of A/D conversion completes for all the selected channels, A/D conversion stops.

If the conversion processes in group A and group B occur at the same time, those conversions cannot be controlled separately. In this case, set group A Priority Control Setting bit (ADGSPCR.PGS) in the A/D Group Scan Priority Control Register (ADGSPCR) to 1 to assign a priority to group A conversion.

In group scan mode, select different channels and triggers for group A and group B.

Only set the ADCS[1:0] bits when the ADST bit is 0. Do not set the ADCS[1:0] bits at the same time that you write 1 to the ADST bit.

**Table 42.19** Selectable targets for A/D conversion depending on scan and double-trigger mode settings

Scan mode setting	Double-trigger mode setting	Targets for A/D conversion				
		Self-diagnosis	Analog input (group A)	Analog input (group B)	Temperature sensor output	Internal reference voltage
Single scan	DBLE = 0	✓	✓	—	✓	✓
	DBLE = 1	—	✓ (1 ch only)	—	—	—
Continuous scan	DBLE = 0	✓	✓	—	✓	✓
	DBLE = 1	—	—	—	—	—
Group scan	DBLE = 0	✓	✓	✓	✓	✓
	DBLE = 1	—	✓ (1 ch only)	✓	✓	✓

Note: ✓: Selectable, —: Not selectable

### ADST bit (A/D Conversion Start)

The ADST bit starts or stops the A/D conversion process. Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and the conversion target analog input.

[Setting conditions]

- 1 is written.
- The synchronous trigger (GPT, ELC) selected in the ADSTRGR.TRSA[5:0] bits is detected when ADCSR.EXTRG is 0 and ADCSR.TRGE is 1.
- The synchronous trigger (GPT, ELC) selected in the ADSTRGR.TRSB[5:0] bits is detected when ADCSR.TRGE is set to 1 in group scan mode.
- The asynchronous trigger is detected when the ADCSR.TRGE and ADCSR.EXTRG bits are set to 1 and the ADSTRGR.TRSA[5:0] bits are set to 0x00.
- When group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), the ADGSPCR.GBRP bit is set to 1, and each time A/D conversion on the group with the lowest priority is started.

[Clearing conditions]

- 0 is written.
- The A/D conversion of all the selected channels, the temperature sensor output the internal reference voltage completes in single scan mode.
- Group A scan completes in group scan mode.
- Group B scan completes in group scan mode.
- When group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), the ADGSPCR.GBRSCN bit is set to 1, and A/D conversion on the group with the lowest priority started by trigger completes.

Note: When group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), do not set the ADST bit to 1.

Note: When group priority operation mode is enabled (ADCSR.ADCS[1:0] = 01b and ADGSPCR.PGS = 1), do not set the ADST bit to 0. When forcing A/D conversion to terminate, follow the procedure for clearing the ADST bit.

### 42.2.9 ADANSA0 : A/D Channel Select Register A0

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSA 15	ANSA 14	ANSA 13	ANSA 12	ANSA 11	ANSA 10	ANSA 09	ANSA 08	ANSA 07	ANSA 06	ANSA 05	ANSA 04	ANSA 03	ANSA 02	ANSA 01	ANSA 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSA15 to ANSA00 <sup>*1</sup>	A/D Conversion Channels Select Bit 15 (ANSA15) is associated with ANm15 and bit 0 (ANSA00) is associated with ANm00. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

Note 1. 00 to 08 (unit 0), 00 to 05 (unit 1) is available.

ADANSA0 register selects analog input channels for A/D conversion. In group scan mode, this register selects group A channels.

Only set the ADANSA0 register when the ADCSR.ADST bit is 0.

#### ANSAn bits (A/D Conversion Channels Select)

The ADANSA0 register selects any combination of analog input channels for A/D conversion. The channels and the number of channels can be arbitrarily set.

In double trigger mode, the channels selected in the ADANSA0 register are invalid, and the channel selected in the ADCSR.DBLANS[4:0] bits is selected in group A instead.

When group scan mode is selected, do not select the channels specified in A/D Channel Select Register B0 (ADANSB0).

VBATT 1/3 voltage monitor output correspond to ADC120.ADANSA0.ANSA03 or ADC121.ADANSA0.ANSA03 bit.

Security attribution function between ADC12n ( $n = 0, 1$ ) and PORT. Detail conversion result please see [section 42.3.12. Security Attribution between ADCn \( \$n = 0, 1\$ \) and PORT \(PmSAR \( \$m = 0\$  to 9\)\)](#).

### 42.2.10 ADANSB0 : A/D Channel Select Register B0

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x014

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ANSB 15	ANSB 14	ANSB 13	ANSB 12	ANSB 11	ANSB 10	ANSB 09	ANSB 08	ANSB 07	ANSB 06	ANSB 05	ANSB 04	ANSB 03	ANSB 02	ANSB 01	ANSB 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ANSB15 to ANSB00 <sup>*1</sup>	A/D Conversion Channels Select Bit 15 (ANSB15) is associated with ANm15 and bit 0 (ANSB00) is associated with ANm00. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

Note 1. 00 to 08 (unit 0), 00 to 05 (unit 1) is available.

ADANSB0 selects analog input channels for A/D conversion in group B when group scan mode is selected. The ADANSB0 register is not used in any scan mode other than group scan mode.

Only set the ADANSB0 register when the ADCSR.ADST bit is 0.

**ANSBn bits (A/D Conversion Channels Select)**

The ADANSB0 register selects any combination of analog input channels in group B for A/D conversion when group scan mode is selected. The ADANSB0 register is used for group scan mode only and not for any other modes.

Do not select channels specified in group A as selected in the ADANSA0 register or the ADCSR.DBLANS[4:0] bits in double-trigger mode.

VBATT 1/3 voltage monitor output correspond to ADC120.ADANSB0.ANSB03 or ADC121.ADANSB0.ANSB03.

Security attribution function between ADC12n (n = 0, 1) and PORT (PmSAR). Detail conversion result please see [section 42.3.12. Security Attribution between ADCn \(n = 0, 1\) and PORT \(PmSAR \(m = 0 to 9\)\)](#).

**42.2.11 ADADS0 : A/D-Converted Value Addition/Average Channel Select Register 0**

Base address: ADC12m = 0x4033\_2000 + 0x0200 × m (m = 0, 1)  
 ADC12m\_NS = 0x5033\_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADS15	ADS14	ADS13	ADS12	ADS11	ADS10	ADS09	ADS08	ADS07	ADS06	ADS05	ADS04	ADS03	ADS02	ADS01	ADS00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADS15 to ADS00*1	A/D-Converted Value Addition/Average Channel Select Bit 15 (ADS15) is associated with ANm15 and bit 0 (ADS00) is associated with ANm00. 0: Do not select associated input channel. 1: Select associated input channel.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

Note 1. 00 to 08 (unit 0), 00 to 05 (unit 1) is available.

**ADSn bits (A/D-Converted Value Addition/Average Channel Select)**

The ADSn bits determine which A/D-converted channels are subject to A/D-converted value addition/averaging. When an ADSn bit associated with a channel selected for A/D conversion is set to 1, A/D conversion of the analog input of the respective channel is performed successively 1, 2, 3, 4, or 16 times, as specified in the ADC[2:0] bits in the ADADC register.

When the ADADC.AVEE bit is 0, the value obtained by addition is stored in the A/D data register. When the ADADC.AVEE bit is 1, the mean value of the results obtained by addition is stored in the A/D data register.

The ADSn bits apply only to channels that are selected for A/D conversion in:

- The ANSAn bits in the ADANSA0 register or the DBLANS[4:0] bits in the ADCSR register
- The ANSBn bits in the ADANSB0 register

For channels on which the A/D conversion is performed and for which addition/average mode is not selected, a normal 1-time conversion is executed, and the conversion result is stored in the A/D data register.

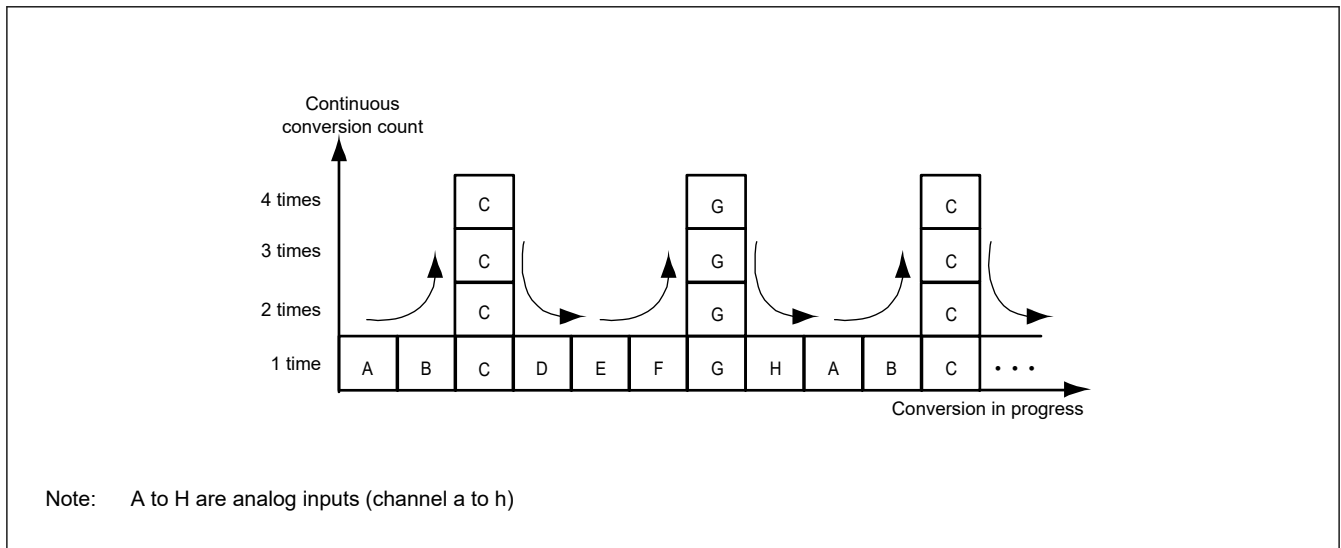
Only set ADADS0 register bits when the ADCSR.ADST bit is 0.

Figure 42.3 shows a scanning operation sequence in which the ADADS0 register bits (channel c and g) are set to 1. In this figure:

- Addition mode is selected (ADADC.AVEE = 0)
- The number of conversions is set to 4 (ADADC.ADC[1:0] = 11b)
- The analog input channels (a to h) are selected by ADANSA0 register in continuous scan mode (ADCSR.ADCS[1:0] = 10b).

The conversion process begins with analog input A (channel a). The analog input C (channel c) conversion is performed successively 4 times and the added value is returned to A/D Data Register c (ADDRc). Next, the analog input D (channel d) conversion process is started. The analog input G (channel g) is performed successively 4 times and the added value is

returned to A/D Data Register g (ADDRg). After conversion of analog input H (channel h), the conversion operation repeats in the same sequence starting with analog input A (channel a).



**Figure 42.3** Scan conversion sequence with ADADC.ADC[2:0] = 011b, set 1 for analog inputs C and G by ADADS0

### 42.2.12 ADADC : A/D-Converted Value Addition/Average Count Select Register

Base address: ADC12m = 0x4033\_2000 + 0x0200 × m (m = 0, 1)  
 ADC12m\_NS = 0x5033\_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x00C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	AVEE	—	—	—	—	ADC[2:0]		

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	ADC[2:0]	Addition/Average Count Select 0 0 0: 1-time conversion (no addition, same as normal conversion) 0 0 1: 2-time conversion (1 addition) 0 1 0: 3-time conversion (2 additions) 0 1 1: 4-time conversion (3 additions) 1 0 1: 16-time conversion (15 additions) Others: Setting prohibited	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	AVEE	Average Mode Select 0: Enable addition mode 1: Enable average mode	R/W

Note: S-TYPE-3, P-TYPE-3

ADADC sets the addition or average mode and addition count for A/D conversion. [Table 42.20](#) lists the settable combinations of ADADC register.

**Table 42.20** Settable combinations of ADADC register

Mode select	Resolution	Conversion time				
		1-time	2-time	3-time	4-time	16-time
Addition mode (AVEE = 0)	8-bit	✓	✓	✓	✓	—
	10-bit	✓	✓	✓	✓	—
	12-bit	✓	✓	✓	✓	✓
Average mode (AVEE = 1)	8, 10, 12 bits	—	✓	—	✓	—

Note: ✓: Selectable, —: Not selectable

### ADC[2:0] bits (Addition/Average Count Select)

The ADC[2:0] bits set the addition count in all channels for which A/D conversion and addition/average mode are selected, including the channel selected in double trigger mode with the ADCSR.DBLANS[4:0] bits. The count also applies to A/D conversion of the temperature sensor output and the internal reference voltage.

When self-diagnosis is executed (ADCER.DIAGM = 1), do not set the ADC[2:0] bits to any value other than 000b.

### AVEE bit (Average Mode Select)

The AVEE bit selects addition or average mode in all channels for which A/D conversion and addition/average mode are selected, including the channels selected in double-trigger mode in the ADCSR.DBLANS[4:0] bits, temperature sensor output, internal reference voltage.

## 42.2.13 ADCER : A/D Control Extended Register

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x00E

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	ADRF MT	—	—	—	DIAG M	DIAGL D	DIAGVAL[1:0]	—	—	ACE	—	—	ADPRC[1:0]	—
------------	------------	---	---	---	-----------	------------	--------------	---	---	-----	---	---	------------	---

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
2:1	ADPRC[1:0]	A/D Conversion Accuracy Specify 0 0: 12-bit accuracy 0 1: 10-bit accuracy 1 0: 8-bit accuracy 1 1: Setting prohibited	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	ACE	A/D Data Register Automatic Clearing Enable 0: Disable automatic clearing 1: Enable automatic clearing	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	DIAGVAL[1:0]	Self-Diagnosis Conversion Voltage Select 0 0: Setting prohibited when self-diagnosis is enabled 0 1: 0 volts 1 0: Reference voltage <sup>*1</sup> × 1/2 1 1: Reference voltage <sup>*1</sup>	R/W
10	DIAGLD	Self-Diagnosis Mode Select 0: Select rotation mode for self-diagnosis voltage 1: Select mixed mode for self-diagnosis voltage	R/W

Bit	Symbol	Function	R/W
11	DIAGM	Self-Diagnosis Enable 0: Disable ADC12 self-diagnosis 1: Enable ADC12 self-diagnosis	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	ADRFMT	A/D Data Register Format Select 0: Select right-justified for the A/D data register format 1: Select left-justified for the A/D data register format	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. The reference voltage refers to VREFH0 for unit 0 and to VREFH for unit 1.

### ADPRC[1:0] bit (A/D Conversion Accuracy Specify)

The ADPRC[1:0] bits set the A/D conversion accuracy. Changing the A/D conversion accuracy also changes the bit width of valid data stored in the result register and the A/D conversion time. For details, see [section 42.3.6. Analog Input Sampling and Scan Conversion Time](#). Only set the ADPRC[1:0] bits while the ADCSR.ADST bit is 0.

### ACE bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0) of the ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR or ADVMDR register after any of these registers is read by the CPU or DTC. Automatic clearing of the A/D data registers enables detection of failures that are not updated in the A/D data registers. For details, see [section 42.3.7. Usage Example of A/D Data Register Automatic Clearing Function](#).

### DIAGVAL[1:0] bits (Self-Diagnosis Conversion Voltage Select)

The DIAGVAL[1:0] bits select the voltage value used in self-diagnosis fixed voltage mode. For details, see the DIAGLD bit description.

Do not execute self-diagnosis by setting the DIAGLD bit to 1 when the DIAGVAL[1:0] bits are set to 00b.

### DIAGLD bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated, or the fixed voltage is used in self-diagnosis.

Setting the DIAGLD bit to 0 selects conversion of the voltages in rotation mode, where 0 V, the reference voltage  $\times 1/2$ , and the reference voltage are converted, in that order. After reset and when self-diagnosis voltage rotation mode is selected, self-diagnosis is executed from 0 V. The self-diagnosis voltage value does not return to 0 V when scan conversion completes. When scan conversion is restarted, rotation starts at the voltage value following the previous value.

Setting the DIAGLD bit to 1 selects fixed voltage, in which the fixed voltage specified in the ADCER.DIAGVAL[1:0] bits is converted. If fixed mode is switched to rotation mode, rotation starts at the fixed voltage value.

Only set the DIAGLD bit when the ADCSR.ADST bit is 0.

### DIAGM bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.

Self-diagnosis is used to detect a failure of the ADC12. In self-diagnosis mode, one of the three voltage values (0 V, the reference voltage  $\times 1/2$ , or the reference voltage) is converted. When conversion completes, information on the converted voltage and the conversion result is stored into the A/D Self-Diagnosis Data Register (ADRD). The ADRD register can be read to determine whether the conversion result falls within the normal or abnormal range.

Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. In double trigger mode (ADCSR.DBLE = 1), self-diagnosis (DIAGM = 0) is deselected. When self-diagnosis is selected in group scan mode, self-diagnosis is executed separately for group A and group B.

Only set the DIAGM bit when the ADCSR.ADST bit is 0.

### ADRFMT bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for data to be stored in the ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADVMDR, ADCMPDR0/1, ADWINLLB, ADWINULB, or ADRD register.

Only set the ADRFMT bit when the ADCSR.ADST bit is 0.



### 42.2.14 ADSTRGR : A/D Conversion Start Trigger Select Register

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x010

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TRSA[5:0]						—	—	TRSB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	TRSB[5:0]	A/D Conversion Start Trigger Select for Group B Select the A/D conversion start trigger for group B in group scan mode.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TRSA[5:0]	A/D Conversion Start Trigger Select Select the A/D conversion start trigger in single scan mode and continuous scan mode. In group scan mode, the A/D conversion start trigger for group A is selected.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

#### TRSB[5:0] bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[5:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[5:0] bits must only be set in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting a software trigger or an asynchronous trigger is prohibited. In group scan mode, set the TRSB[5:0] bits to a value other than 0x00 and set the ADCSR.TRGE bit to 1.

When group A is given priority in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 0x3F. The issuance period for a conversion trigger must be more than or equal to the actual scan conversion time ( $t_{SCAN}$ ). If the issuance period is less than  $t_{SCAN}$ , A/D conversion by the trigger might have no effect.

When the GPT module is selected as an A/D conversion start trigger, a delay for synchronization processing occurs. For details, see [section 42.3.6. Analog Input Sampling and Scan Conversion Time](#).

[Table 42.21](#) lists the A/D conversion startup sources selected in the TRSB[5:0] bits.

**Table 42.21 Selection of A/D conversion start sources in the TRSB[5:0] bits (1 of 2)**

Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source deselected state	—	1	1	1	1	1	1
ELC_AD00 (unit 0) and ELC_AD10 (unit 1)	ELC	0	0	1	0	0	1
ELC_AD01 (unit 0) and ELC_AD11 (unit 1)	ELC	0	0	1	0	1	0
ELC_AD00 (unit 0) and ELC_AD10 (unit 1), ELC_AD01 (unit 0) and ELC_AD11 (unit 1)	ELC	0	0	1	0	1	1
GTCIADA0 (unit 0) and GTCIADA4 (unit 1)	GPT	0	1	0	0	0	1
GTCIADB0 (unit 0) and GTCIADB4 (unit 1)	GPT	0	1	0	0	1	0
GTCIADA1 (unit 0) and GTCIADA5 (unit 1)	GPT	0	1	0	0	1	1
GTCIADB1 (unit 0) and GTCIADB5 (unit 1)	GPT	0	1	0	1	0	0

**Table 42.21 Selection of A/D conversion start sources in the TRSB[5:0] bits (2 of 2)**

Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
GTCIADA2 (unit 0)	GPT	0	1	0	1	0	1
GTCIADB2 (unit 0)	GPT	0	1	0	1	1	0
GTCIADA3 (unit 0)	GPT	0	1	0	1	1	1
GTCIADB3 (unit 0)	GPT	0	1	1	0	0	0
GTCIADA0 (unit 0) and GTCIADA4 (unit 1), GTCIADB0 (unit 0) and GTCIADB4 (unit 1)	GPT	0	1	1	0	0	1
GTCIADA1 (unit 0) and GTCIADA5 (unit 1), GTCIADB1 (unit 0) and GTCIADB5 (unit 1)	GPT	0	1	1	0	1	0
GTCIADA2 (unit 0), GTCIADB2 (unit 0)	GPT	0	1	1	0	1	1
GTCIADA3 (unit 0), GTCIADB3 (unit 0)	GPT	0	1	1	1	0	0

**TRSA[5:0] bits (A/D Conversion Start Trigger Select)**

The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode, or the trigger to start scanning of group A analog inputs in group scan mode. When scanning is executed in group scan mode or double trigger mode, software trigger or asynchronous trigger is prohibited.

- When using a synchronous trigger (GPT, ELC), set the TRGE bit in the ADCSR register to 1 and set the EXTRG bit in the ADCSR register to 0.
- When using the asynchronous trigger (ADTRGn (n = 0, 1)), set the TRGE bit in the ADCSR register to 1 and set the EXTRG bit in the ADCSR register to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit, the ADCSR.EXTRG bit, or the TRSA[5:0] bits.

The issuance period for a conversion trigger must be more than or equal to the actual scan conversion time (tSCAN). If the issuance period is less than tSCAN, A/D conversion by a trigger might have no effect.

When the GPT module is selected as an A/D conversion start trigger, a delay for synchronization processing occurs. For details, see [section 42.3.6. Analog Input Sampling and Scan Conversion Time](#).

[Table 42.22](#) lists the A/D conversion start sources selected in the TRSA[5:0] bits.

**Table 42.22 Selection of A/D activation sources in the TRSA[5:0] bits (1 of 2)**

Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source deselected state	—	1	1	1	1	1	1
ADTRGn (n = 0, 1)	Input pin for the trigger	0	0	0	0	0	0
ELC_AD00 (unit 0) and ELC_AD10 (unit 1)	ELC	0	0	1	0	0	1
ELC_AD01 (unit 0) and ELC_AD11 (unit 1)	ELC	0	0	1	0	1	0
ELC_AD00 (unit 0) and ELC_AD10 (unit 1), ELC_AD01 (unit 0) and ELC_AD11 (unit 1)	ELC	0	0	1	0	1	1
GTCIADA0 (unit 0) and GTCIADA4 (unit 1)	GPT	0	1	0	0	0	1

**Table 42.22 Selection of A/D activation sources in the TRSA[5:0] bits (2 of 2)**

Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
GTCIADB0 (unit 0) and GTCIADB4 (unit 1)	GPT	0	1	0	0	1	0
GTCIADA1 (unit 0) and GTCIADA5 (unit 1)	GPT	0	1	0	0	1	1
GTCIADB1 (unit 0) and GTCIADB5 (unit 1)	GPT	0	1	0	1	0	0
GTCIADA2 (unit 0)	GPT	0	1	0	1	0	1
GTCIADB2 (unit 0)	GPT	0	1	0	1	1	0
GTCIADA3 (unit 0)	GPT	0	1	0	1	1	1
GTCIADB3 (unit 0)	GPT	0	1	1	0	0	0
GTCIADA0 (unit 0) and GTCIADA4 (unit 1), GTCIADB0 (unit 0) and GTCIADB4 (unit 1)	GPT	0	1	1	0	0	1
GTCIADA1 (unit 0) and GTCIADA5 (unit 1), GTCIADB1 (unit 0) and GTCIADB5 (unit 1)	GPT	0	1	1	0	1	0
GTCIADA2 (unit 0), GTCIADB2 (unit 0)	GPT	0	1	1	0	1	1
GTCIADA3 (unit 0), GTCIADB3 (unit 0)	GPT	0	1	1	1	0	0

### 42.2.15 ADEXICR : A/D Conversion Extended Input Control Registers

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x012

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	OCSB	TSSB	OCSA	TSSA	—	—	—	—	—	—	OCSA D	TSSA D
------------	---	---	---	---	------	------	------	------	---	---	---	---	---	---	-----------	-----------

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	TSSAD	Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select 0: Do not select addition/average mode for temperature sensor output. 1: Select addition/average mode for temperature sensor output.	R/W
1	OCSAD	Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select 0: Do not select addition/average mode for internal reference voltage. 1: Select addition/average mode for internal reference voltage.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	TSSA	Temperature Sensor Output A/D Conversion Select 0: Disable A/D conversion of temperature sensor output 1: Enable A/D conversion of temperature sensor output	R/W
9	OCSA	Internal Reference Voltage A/D Conversion Select 0: Disable A/D conversion of internal reference voltage 1: Enable A/D conversion of internal reference voltage	R/W
10	TSSB	Temperature Sensor Output A/D Conversion Select for Group B 0: Disable A/D conversion of temperature sensor output 1: Enable A/D conversion of temperature sensor output	R/W

Bit	Symbol	Function	R/W
11	OCSB	Internal Reference Voltage A/D Conversion Select for Group B 0: Disable A/D conversion of internal reference voltage 1: Enable A/D conversion of internal reference voltage	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

#### **TSSAD bit (Temperature Sensor Output A/D-Converted Value Addition/Average Mode Select)**

When the TSSAD bit is set to 1, A/D conversion of the temperature sensor output is selected and performed successively the number of times specified in the ADC[2:0] bits in ADADC. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is returned to the A/D Temperature Sensor Data Register (ADTSDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADTSDR.

Only set the TSSAD bit while the ADCSR.ADST bit is 0.

#### **OCSAD bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)**

When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is selected and performed successively the number of times specified in the ADC[2:0] bits in ADADC. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is returned to the A/D Internal Reference Voltage Data Register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is returned to ADOCDR.

Only set the OCSAD bit while the ADCSR.ADST bit is 0.

#### **TSSA bit (Temperature Sensor Output A/D Conversion Select)**

The TSSA bit selects A/D conversion of the temperature sensor output for group A in single scan mode, continuous scan mode, or group scan mode. When A/D conversion of the temperature sensor output is selected and performed, set the ADCSR.DBLE bit to 0.

Only set the TSSA bit while the ADCSR.ADST bit is 0.

#### **OCSA bit (Internal Reference Voltage A/D Conversion Select)**

The OCSA bit selects A/D conversion of the internal reference voltage for group A in single scan mode, continuous scan mode, or group scan mode. When A/D conversion of the internal reference voltage is selected and performed, set the ADCSR.DBLE bit to 0.

Only set the OCSA bit while the ADCSR.ADST bit is 0. In addition, wait for 400 ns or more after the OCSA bit is set to 1 before starting A/D conversion.

#### **TSSB bit (Temperature Sensor Output A/D Conversion Select for Group B)**

The TSSB bit selects A/D conversion of the temperature sensor output for group B in group scan mode. Only set the TSSB bit while the ADCSR.ADST bit is 0. Do not set the TSSB bit to 1 while the TSSA bit is 1.

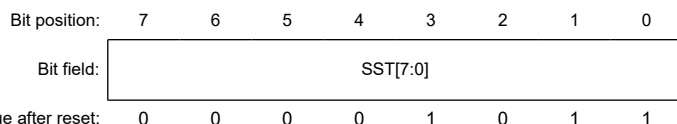
#### **OCSB bit (Internal Reference Voltage A/D Conversion Select for Group B)**

The OCSB bit selects A/D conversion of the internal reference voltage for group B in group scan mode. Only set the OCSB bit while the ADCSR.ADST bit is 0. Do not set the OCSB bit to 1 while the OCSA bit is 1. Moreover, start the A/D conversion after waiting for 400 ns or more after the OCSB bit is set to 1.

### 42.2.16 ADSSTRn/ADSSTRT/ADSSTRO/ADSSTRV: A/D Sampling State Register (n = 0 to 8)

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  (m = 0, 1)  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  (m = 0, 1)

Offset address:  $0x0E0 + 0x1 \times n$  (n = 0 to 8, m = 0)  
 $0x0E0 + 0x1 \times n$  (n = 0 to 5, m = 1)  
 0x0DE (ADSSTRT)  
 0x0DF (ADSSTRO)  
 0x0E3 (ADSSTRV)



Bit	Symbol	Function	R/W
7:0	SST[7:0]	Sampling Time Setting These bits set the sampling time in the range from 5 to 255 states.	R/W

Note: S-TYPE-3, P-TYPE-3

The ADSSTRn register sets the sampling time for analog input.

The sampling time can be adjusted if the impedance of the analog input signal source is too high to secure sufficient sampling time, or if the ADCLK clock is slow. The set value indicates the time for one ADCLK cycle, and the required sampling time is specified by the voltage conditions. For details, see [section 52.5. ADC12 Characteristics](#).

ADSSTRV register is an alias for the ADSSTR3 register.

The lower limit of the sampling time setting depends on the frequency ratio:

- If the frequency ratio of PCLKA to PCLKC (ADCLK) = 1:1, 2:1, 4:1, or 8:1 the sampling time must be set to a value of more than 5 states
- If the frequency ratio of PCLKA to PCLKC (ADCLK) = 1:2 or 1:4, the sampling time must be set to a value of more than 6 states.

[Table 42.23](#) shows the relationship between the A/D Sampling State Register and the associated channels. For details, see [section 42.3.6. Analog Input Sampling and Scan Conversion Time](#).

Only set the SST[7:0] bits when the ADCSR.ADST bit is 0.

**Table 42.23 Relationship between A/D sampling state register and associated channels**

Bit name	Associated channels
ADSSTRn.SST[7:0] bits (n = 0 to 8 (unit 0), 0 to 5 (unit 1))*1	AN000 to AN008, AN100 to AN105
ADSSTRT.SST[7:0] bits	Temperature sensor output
ADSSTRO.SST[7:0] bits	Internal reference voltage
ADSSTRV.SST[7:0] bits	VBATT 1/3 voltage monitor output*2

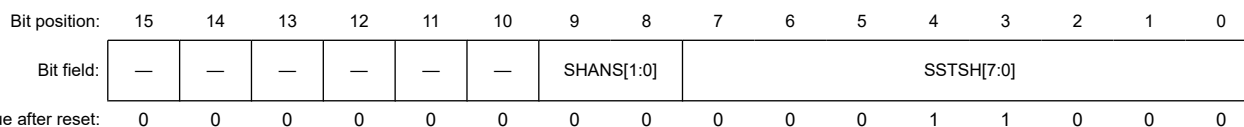
Note 1. When the self-diagnosis function is selected, the sampling time set in the ADSSTR0.SST[7:0] bits is applied.

Note 2. For the stabilization time, see [section 11, Battery Backup Function](#).

### 42.2.17 ADShCR : A/D Sample and Hold Circuit Control Register

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  (m = 0)  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  (m = 0)

Offset address: 0x066



Bit	Symbol	Function	R/W
7:0	SSTSH[7:0]	Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting Sampling time (4 to 255 states).	R/W
9:8	SHANS[1:0]	Channel-Dedicated Sample-and-Hold Circuit Bypass Select Select whether to use or bypass channel-dedicated sample-and-hold circuits for AN000, AN001 (unit 0). 0: Bypass the circuits 1: Use the circuits	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

### SSTSH[7:0] bits (Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting)

The SSTSH[7:0] bits set the sampling time for the channel-dedicated sample-and-hold circuits. If one state is 1 ADCLK (A/D conversion clock) cycle and the ADCLK clock is 60 MHz, one state is 16.7 ns. The initial value is 24 states. The sampling time can be adjusted if the impedance of the analog input signal source is too high to secure sufficient sampling time, or if the ADCLK clock is slow.

Only set the SSTSH[7:0] bits while the ADCSR.ADST bit is 0. The sampling time must be set to a value that is 4 states or more and 255 or less.

### SHANS[1:0] bits (Channel-Dedicated Sample-and-Hold Circuit Bypass Select)

The SHANS[1:0] bits select whether to use or bypass the channel-dedicated sample-and-hold circuits for AN000, AN001 (unit 0). In unit 0, the SHANS[0] bit is associated with AN000, the SHANS[1] bit with AN001.

If any channel from among AN000, AN001 (unit 0) is selected for group B while operation is in group scan mode under group A priority control, use this setting to bypass the dedicated sample-and-hold circuit of the channel.

Only set the SHANS[1:0] bits while the ADCSR.ADST bit is 0 and the ADSHMSR.SHMD bit is 0.

## 42.2.18 ADSHMSR : A/D Sample and Hold Operation Mode Selection Register

Base address: ADC12m = 0x4033\_2000 + 0x0200 × m (m = 0)  
ADC12m\_NS = 0x5033\_2000 + 0x0200 × m (m = 0)

Offset address: 0x07C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SHMD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SHMD	Sampling Operation Selection 0: Sampling Operation Selection 1: Enable continuous sampling function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

### SHMD bit (Sampling Operation Selection)

Setting SHMD to 1 enables the constant sampling function of the channel-dedicated sample-and-hold selected in the ADSHCR.SHANS[1:0] bits. Only set the SHMD bit while the ADCSR.ADST bit is 0.

When the sampling function is enabled, the sample-and-hold circuit operates sampling while the ADC12 is not operating, and it operates holding while the ADC12 is operating.

Note: The ADCSR.ADST bit must become 1 after a time of 400 ns or more elapses after the SHMD bit is set to 1 (when the permissible signal source impedance is 1 kΩ). The sampling period of the sample-and-hold circuit must be 400 ns or more (when the permissible signal source impedance is 1 kΩ).

### 42.2.19 ADDISCR : A/D Disconnection Detection Control Register

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x07A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PCHG	ADNDIS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ADNDIS[3:0]	Disconnection Detection Assist Setting 0x0: The disconnection detection assist function is disabled 0x1: Setting prohibited Others: The number of states for the discharge or precharge period.	R/W
4	PCHG	Precharge/discharge select 0: Discharge 1: Precharge	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The ADDISCR register selects either precharge or discharge, and the period of precharge or discharge for the A/D disconnection detection assist function. Only set the ADDISCR register when the ADCSR.ADST bit is 0. When the temperature sensor output or internal reference voltage is converted, the A/D converter executes discharge automatically.

Disable the disconnection detection assist function if any of the following functions are used:

- The temperature sensor
- The internal reference voltage
- A/D self-diagnosis
- VBATT 1/3 voltage monitor

#### ADNDIS[3:0] bits (Disconnection Detection Assist Setting)

The ADNDIS[3:0] bits specify the period of precharge or discharge. When ADNDIS[3:0] = 0000b, the disconnection detection assist function is disabled. Setting the ADNDIS[3:0] bits to 0001b is prohibited. Except when ADNDIS[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge or discharge. When the ADNDIS[3:0] bits are set to any values other than 0000b or 0001b, the disconnection detection assistance function is enabled. The ADNDIS[4:0] should be set while the ADCSR.ADST bit is 0. When the ADNDIS[3:0] bits are set to any values other than 0000b, and the disconnection detection assistance is enabled, the disconnection detection assistance for the channel-dedicated sample-and-hold circuit used for analog inputs are also enabled. When the temperature sensor output or internal reference voltage is converted or self-diagnosis is used, the disconnection detection assistance cannot be used. In that case, the ADNDIS[3:0] bits should be set to 0000b.

#### PCHG bit (Precharge/discharge select)

The PCHG bit selects either precharge or discharge.

### 42.2.20 ADGSPCR : A/D Group Scan Priority Control Register

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x080

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GBRP	LGRR S	—	—	—	—	—	—	—	—	—	—	—	—	GBRS CN	PGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PGS	Group Priority Operation Setting* <sup>1</sup> 0: Operate without group priority control. 1: Operate with group priority control.	R/W
1	GBRSCN	Lower-Priority Group Restart Setting (enabled only when PGS = 1 and reserved when PGS = 0.) 0: Disable rescanning of the group that was stopped in group priority operation 1: Enable rescanning of the group that was stopped in group priority operation.	R/W
13:2	—	These bits are read as 0. The write value should be 0.	R/W
14	LGRRS	Restart Channel Select Enabled only when PGS = 1 and GBRSCN = 1. 0: Start rescanning from the first channel for scanning 1: Start rescanning from the channel for which A/D conversion is not completed.	R/W
15	GBRP	Single Scan Continuous Start* <sup>2</sup> (enabled only when PGS = 1 and reserved when PGS = 0.) 0: Single scan is not continuously activated. 1: Single scan for the group with the lower-priority is continuously activated.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. The ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode) before setting PGS to 1. Operation is not guaranteed if these bits are set to any other value.

Note 2. When the GBRP bit is set to 1, single scan is performed continuously for the group with the lower-priority regardless of the setting in the GBRSCN bit.

### PGS bit (Group Priority Operation Setting)

The PGS bit controls group priority operation in group scan mode. Set the PGS bit to 1 to enable group priority operation.

The ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode) before setting the PGS bit to 1. Operation is not guaranteed if the bits are set to any other value.

When the PGS bit is set to 0, a clear operation must be performed by software as described in [section 42.6.3. Constraints on Stopping A/D Conversion](#). When the PGS bit is set to 1, use the settings described in [section 42.3.4.3. Group Priority Operation](#).

### GBRSCN bit (Lower-Priority Group Restart Setting)

The GBRSCN bit controls the restarting of scan operation in group priority operation.

When the GBRSCN bit is set to 1, if the scan operation of a lower-priority group is stopped by a trigger input of a priority group, the lower-priority group scanning is restarted on completion of the priority group scanning. If a trigger of a lower-priority group is input during scanning of the priority group, the lower-priority group scanning is started on completion of the priority group scanning.

When the GBRSCN bit is set to 0, triggers input during scanning are ignored. Set the GBRSCN bit while the ADCSR.ADST bit is 0.

### LGRRS bit (Restart Channel Select)

This bit sets the channel from which rescanning is to be started in group priority operation. The setting of the LGRRS bit is valid when the PGS and GBRSCN bits are 1.

If the LGRRS bit is 0, scanning of a lower-priority group that was stopped in group priority operation is restarted from the first channel after scanning of the priority group completes.

If the LGRRS bit is 1, scanning of a lower-priority group that was stopped in group priority operation is restarted (upon completion of scanning of the priority group) from the channel for which A/D conversion is not complete. If A/D conversion of the addition setting channel was not completed the specified number of times when scanning stopped, A/D conversion of the addition setting channel is performed again the specified number of times when scanning restarts.

Set the LGRRS bit while the ADCSR.ADST bit is 0.

### GBRP bit (Single Scan Continuous Start)

The GBRP bit is set when a single scan operation is to be performed continuously on the group with the lower-priority.

Setting the GBRP bit to 1 starts a single scan of the group with the lower-priority. On completion of the scan, another single scan of the group with the lower-priority is started automatically. If scanning has been stopped during group priority



operation, single scan of the group with the lower-priority is automatically restarted on completion of the A/D conversion of the priority group.

Before setting the GBRP bit to 1, disable input of a trigger for the lower-priority group. If the GBRP bit is set to 1, rescanning is performed only on the group with the lower-priority even if the GBRSCN bit is set to 0.

### 42.2.21 ADCMPCR : A/D Compare Function Control Register

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x090

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPAIE	WCMP E	CMPBIE	—	CMPAE	—	CMPBE	—	—	—	—	—	—	—	CMPAB[1:0]	

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
1:0	CMPAB[1:0]	Window A/B Composite Conditions Setting These bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1). 0 0: Output ADC12i_WCMPM (i = 0, 1) when window A OR window B comparison conditions are met. Otherwise, output ADC12i_WCMPUM (i = 0, 1). 0 1: Output ADC12i_WCMPM (i = 0, 1) when window A EXOR window B comparison conditions are met. Otherwise, output ADC12i_WCMPUM (i = 0, 1). 1 0: Output ADC12i_WCMPM (i = 0, 1) when window A AND window B comparison conditions are met. Otherwise, output ADC12i_WCMPUM (i = 0, 1). 1 1: Setting prohibited.	R/W
8:2	—	These bits are read as 0. The write value should be 0.	R/W
9	CMPBE	Compare Window B Operation Enable 0: Disable compare window B operation. Disable ADC12i_WCMPM (i = 0, 1) and ADC12i_WCMPUM (i = 0, 1) outputs. 1: Enable compare window B operation.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	CMPAE	Compare Window A Operation Enable 0: Disable compare window A operation. Disable ADC12i_WCMPM (i = 0, 1) and ADC12i_WCMPUM (i = 0, 1) outputs. 1: Enable compare window A operation.	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	CMPBIE	Compare B Interrupt Enable 0: Disable ADC12i_CMPBI (i = 0, 1) interrupt when comparison conditions (window B) are met. 1: Enable ADC12i_CMPBI (i = 0, 1) interrupt when comparison conditions (window B) are met.	R/W
14	WCMP E	Window Function Setting 0: Disable window function Window A and window B operate as a comparator to compare the single value on the lower side with the A/D conversion result. 1: Enable window function Window A and window B operate as a comparator to compare the two values on the upper and lower sides with the A/D conversion result.	R/W
15	CMPAIE	Compare A Interrupt Enable 0: Disable ADC12i_CMPAI (i = 0, 1) interrupt when comparison conditions (window A) are met. 1: Enable ADC12i_CMPAI (i = 0, 1) interrupt when comparison conditions (window A) are met.	R/W

Note: S-TYPE-3, P-TYPE-3

**CMPAB[1:0] bits (Window A/B Composite Conditions Setting)**

The CMPAB[1:0] bits are valid when both window A and window B are enabled (CMPAE = 1 and CMPBE = 1) in single scan mode. These bits specify the compare function match/mismatch event output conditions and monitoring conditions of ADWINMON.MONCOMB. Only set the CMPAB[1:0] bits while the ADCSR.ADST bit is 0.

**CMPBE bit (Compare Window B Operation Enable)**

The CMPBE bit enables or disables the compare window B operation. Set the CMPBE bit while the ADCSR.ADST bit is 0. Set this bit to 0 before setting the following registers:

- A/D Channel Select Registers A0, B0 (ADANSA0, ADANSB0)
- OCSB, TSSB, OCSA, or TSSA bits in the A/D Conversion Extended Input Control Register (ADEXICR)
- CMPCHB[5:0] bits in the Window B Channel Select Register (ADCMPBNSR)

**CMPAE bit (Compare Window A Operation Enable)**

The CMPAE bit enables or disables the compare window A operation. Set the CMPAE bit while the ADCSR.ADST bit is 0. Set this bit to 0 before setting the following registers:

- A/D Channel Select Registers A0, B0 (ADANSA0, ADANSB0)
- OCSB, TSSB, OCSA, or TSSA bits in the A/D Conversion Extended Input Control Register (ADEXICR)
- Window A Channel Select Registers 0 (ADCMPANSR0)
- Window A Extended Input Select Register (ADCMPANSER)

**CMPBIE bit (Compare B Interrupt Enable)**

The CMPBIE bit enables or disables the ADC12i\_CMPBI (i = 0, 1) interrupt output when the comparison conditions (window B) are met.

**WCMPE bit (Window Function Setting)**

The WCMPE bit enables or disables the window function. Set the WCMPE bit while the ADCSR.ADST bit is 0.

**CMPAIE bit (Compare A Interrupt Enable)**

The CMPAIE bit enables or disables the ADC12i\_CMPAI (i = 0, 1) interrupt output when the comparison conditions (window A) are met.

**42.2.22 ADCMPANSR0 : A/D Compare Function Window A Channel Select Register 0**

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  (m = 0, 1)  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  (m = 0, 1)

Offset address: 0x094

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPC HA15	CMPC HA14	CMPC HA13	CMPC HA12	CMPC HA11	CMPC HA10	CMPC HA09	CMPC HA08	CMPC HA07	CMPC HA06	CMPC HA05	CMPC HA04	CMPC HA03	CMPC HA02	CMPC HA01	CMPC HA00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPCHA15 to CMPCHA00*1	Compare Window A Channel Select Bit 15 (CMPCHA15) is associated with ANm15 and bit 0 (CMPCHA00) is associated with ANm00. 0: Disable compare function for associated input channel 1: Enable compare function for associated input channel	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

Note 1. 00 to 08 (unit 0), 00 to 05 (unit 1) is available.

**CMPCHAN bits (Compare Window A Channel Select)**

The compare function is enabled by writing 1 to the CMPCHAN bits with the same number as the A/D conversion channel selected in the ADANSA0.ANSAn bits and the ADANSB0.ANSBn bits.

Set the CMPCHAN bits while the ADCSR.ADST bit is 0.

**42.2.23 ADCMPANSER : A/D Compare Function Window A Extended Input Select Register**

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x092

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CMPO CA	CMPT SA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPTSA	Temperature Sensor Output Compare Select 0: Exclude the temperature sensor output from the compare Window A target range. 1: Include the temperature sensor output in the compare Window A target range.	R/W
1	CMPOCA	Internal Reference Voltage Compare Select 0: Exclude the internal reference voltage from the compare Window A target range. 1: Include the internal reference voltage in the compare Window A target range.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

**CMPTSA bit (Temperature Sensor Output Compare Select)**

The compare Window A function is enabled by setting the CMPTSA bit to 1 while the ADEXICR.TSSA bit or the ADEXICR.TSSB bit is 1. Set the CMPTSA bit while the ADCSR.ADST bit is 0.

**CMPOCA bit (Internal Reference Voltage Compare Select)**

The compare window A function is enabled by setting the CMPOCA bit to 1 when the ADEXICR.OCSA and ADEXICR.OCSB bit is 1. Set the CMPOCA bit when the ADCSR.ADST bit is 0.

**42.2.24 ADCMPLR0 : A/D Compare Function Window A Comparison Condition Setting Register 0**

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x098

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPL CHA1 5	CMPL CHA1 4	CMPL CHA1 3	CMPL CHA1 2	CMPL CHA1 1	CMPL CHA1 0	CMPL CHA0 9	CMPL CHA0 8	CMPL CHA0 7	CMPL CHA0 6	CMPL CHA0 5	CMPL CHA0 4	CMPL CHA0 3	CMPL CHA0 2	CMPL CHA0 1	CMPL CHA0 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPLCHA15 to CMPLCHA00 <sup>*1</sup>	<p>Compare Window A Comparison Condition Select</p> <p>These bits set comparison conditions for channels to which Window A comparison conditions are applied.</p> <p>Bit 15 (CMPLCHA15) is associated with ANm15 and bit 0 (CMPLCHA00) is associated with ANm00.</p> <p>Comparison conditions are shown in <a href="#">Figure 42.4</a>.</p> <p>0: When window function is disabled (ADCMPCR.WCMPE = 0):            ADCMPDR0 value &gt; A/D-converted value            When window function is enabled (ADCMPCR.WCMPE = 1):            A/D-converted value &lt; ADCMPDR0 value,            or ADCMPDR1 value &lt; A/D-converted value</p> <p>1: When window function is disabled (ADCMPCR.WCMPE = 0):            ADCMPDR0 value &lt; A/D-converted value            When window function is enabled (ADCMPCR.WCMPE = 1):            ADCMPDR0 value &lt; A/D-converted value &lt; ADCMPDR1 value</p>	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

Note 1. 00 to 08 (unit 0), 00 to 05 (unit 1) is available.

### CMPLCHAN bits (Compare Window A Comparison Condition Select)

The CMPLCHAN bits specify the comparison conditions for channels to which Window A comparison conditions are applied. These bits can be set for each analog input to be compared. When the comparison result of each analog input meets the set condition, the ADCMPDR0.CMPSTCHAN flag sets to 1 and a compare interrupt (ADC12i\_CMPAI (i = 0, 1)) is generated.

Comparison conditions when the window function is disabled			
CMPLCHAN = 0		CMPLCHAN = 1	
ADCMPDR0 value ≤ A/D converted value	Not met	ADCMPDR0 value < A/D converted value	Met
ADCMPDR0 value > A/D converted value	Met	ADCMPDR0 value ≥ A/D converted value	Not met
Comparison conditions when the window function is enabled			
CMPLCHAN = 0			
ADCMPDR1 value < A/D converted value		Met	
ADCMPDR0 value ≤ A/D converted value ≤ ADCMPDR1 value		Not met	
A/D converted value < ADCMPDR0 value		Met	
CMPLCHAN = 1			
ADCMPDR1 value ≤ A/D converted value		Not met	
ADCMPDR0 value < A/D converted value < ADCMPDR1 value		Met	
A/D converted value ≤ ADCMPDR0 value		Not met	

Figure 42.4 Explanation of comparison conditions for compare function Window A

### 42.2.25 ADCMPLER : A/D Compare Function Window A Extended Input Comparison Condition Setting Register

Base address: ADC12m = 0x4033\_2000 + 0x0200 × m (m = 0, 1)  
 ADC12m\_NS = 0x5033\_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x093

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CMPL OCA	CMPL TSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPLTSA	Compare Window A Temperature Sensor Output Comparison Condition Select Comparison conditions are shown in <a href="#">Figure 42.4</a> . 0: When window function is disabled (ADCMPCR.WCMPE = 0) : ADCMPDR0 value > A/D-converted value Compare Window A Temperature Sensor Output Comparison Condition Select When window function is enabled (ADCMPCR.WCMPE = 1) : Compare Window A Temperature Sensor Output Comparison Condition A/D-converted value < ADCMPDR0 value, or A/D-converted value > ADCMPDR1 value 1: When window function is disabled (ADCMPCR.WCMPE = 0) : ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1) : ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W
1	CMPLOCA	Compare Window A Internal Reference Voltage Comparison Condition Select Comparison conditions are shown in <a href="#">Figure 42.4</a> . 0: When window function is disabled (ADCMPCR.WCMPE = 0) : ADCMPDR0 value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADCMPDR0 value, or A/D-converted value > ADCMPDR1 value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADCMPDR0 value < A/D-converted value < ADCMPDR1 value	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

**CMPLTSA bit (Compare Window A Temperature Sensor Output Comparison Condition Select)**

The CMPLTSA bit specifies comparison conditions when the temperature sensor output is the target for the Window A comparison condition. When the temperature sensor output comparison result meets the set condition, the ADCMPSESR.CMPSTTSA flag sets to 1 and a compare interrupt (ADC12i\_CMPAI (i = 0, 1)) is generated.

**CMPLOCA bit (Compare Window A Internal Reference Voltage Comparison Condition Select)**

The CMPLOCA bit specifies comparison conditions when the internal reference voltage is the target for the Window A comparison condition. When the internal reference voltage comparison result meets the set condition, the ADCMPSESR.CMPSTOCA flag sets to 1 and a compare interrupt (ADC12i\_CMPAI (i = 0, 1)) is generated.

**42.2.26 ADCMPDRn : A/D Compare Function Window A Lower-Side/Upper-Side Level Setting Register (n = 0, 1)**

Base address: ADC12m = 0x4033\_2000 + 0x0200 × m (m = 0, 1)  
 ADC12m\_NS = 0x5033\_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x09C + (0x2 × n)

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Note: S-TYPE-3, P-TYPE-3

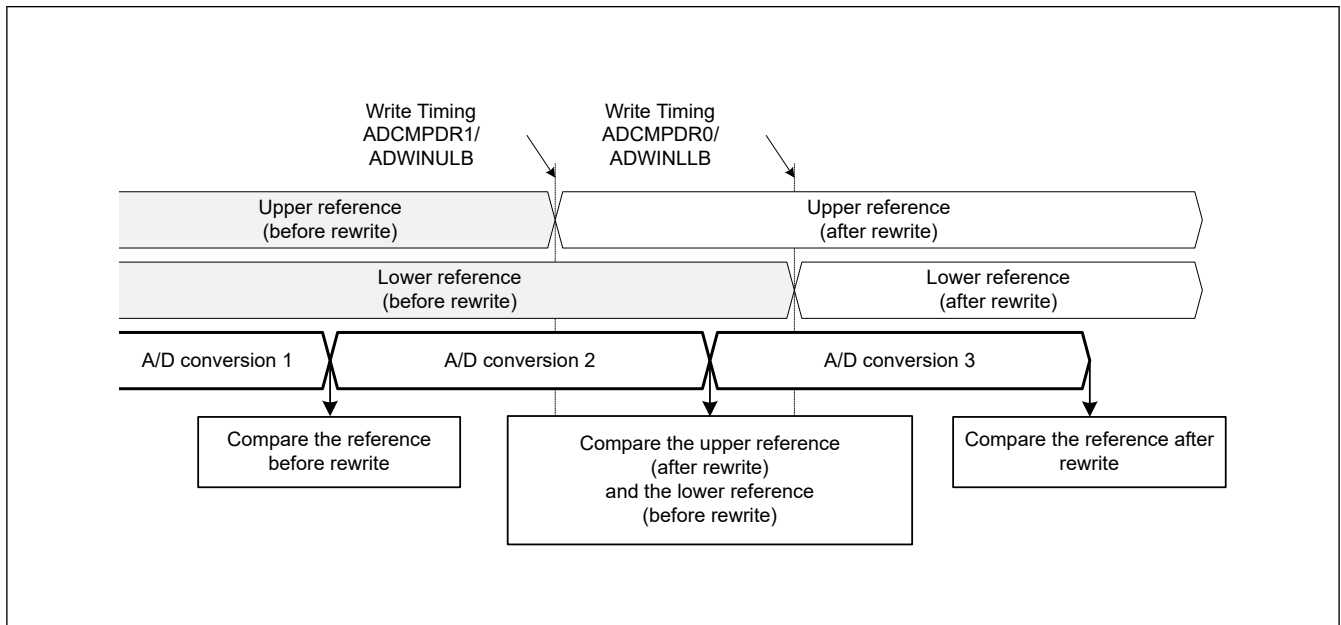
The ADCMPDRy (y = 0, 1) register specifies the reference data when the compare window A function is used. ADCMPDR0 sets the lower reference for window A, and ADCMPDR1 sets the upper reference for window A.

ADCMPDRy are read/write registers.

ADCMPDRy are writable even during A/D conversion. The reference data can be dynamically changed by rewriting register values during A/D conversion\*1.

Set these registers so that the upper reference is not less than the lower reference (ADCMPDR1 ≥ ADCMPDR0). ADCMPDR1 are not used when the window function is disabled.

Note 1. The lower and the upper references are changed when each register is written. For example, when the upper reference value is changed and the lower reference value is being changed, the MCU compares the upper reference (after rewrite), and the lower reference (before rewrite) with the A/D conversion result. See [Figure 42.5](#). If the comparison during the rewriting of these two references is erroneous, then rewrite these reference values when both ADCSR.ADST and the target Compare Window Operation Enable bit (ADCMPCR.CMPAE or ADCMPCR.CMPBE) are 0.



**Figure 42.5 Comparison between upper and lower references before and after a rewrite**

The ADCMPDRy registers use different formats depending on the following conditions:

- The value of A/D Data Register Format Select bit (flush-right or flush-left)
- The value of the A/D Conversion Accuracy Select bit (12-bit, 10-bit, 8-bit)
- The value of A/D-Converted Value Addition/Average Channel Select bits (A/D-converted value addition mode selected or not selected).

The data formats for each condition are shown as follows:

1. When A/D-converted value addition mode is not selected
  - Flush-right data with 12-bit accuracy — Lower 12 bits ([11:0]) are valid
  - Flush-right data with 10-bit accuracy — Lower 10 bits ([9:0]) are valid
  - Flush-right data with 8-bit accuracy — Lower 8 bits ([7:0]) are valid
  - Flush-left data with 12-bit accuracy — Upper 12 bits ([15:4]) are valid
  - Flush-left data with 10-bit accuracy — Upper 10 bits ([15:6]) are valid
  - Flush-left data with 8-bit accuracy — Upper 8 bits ([15:8]) are valid
2. When A/D-converted value addition mode is selected
  - Flush-right data with 12-bit accuracy — Lower 14 bits ([13:0]) are valid
  - Flush-right data with 10-bit accuracy — Lower 12 bits ([11:0]) are valid
  - Flush-right data with 8-bit accuracy — Lower 10 bits ([9:0]) are valid
  - Flush-left data with 12-bit accuracy — Upper 14 bits ([15:2]) are valid
  - Flush-left data with 10-bit accuracy — Upper 12 bits ([15:4]) are valid
  - Flush-left data with 8-bit accuracy — Upper 10 bits ([15:6]) are valid

### 42.2.27 ADWINnLB : A/D Compare Function Window B Lower-Side/Upper-Side Level Setting Register (n = L, U)

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  (m = 0, 1)  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  (m = 0, 1)

Offset address: 0x0A8 (n = L)  
 0x0AA (n = U)



Note: S-TYPE-3, P-TYPE-3

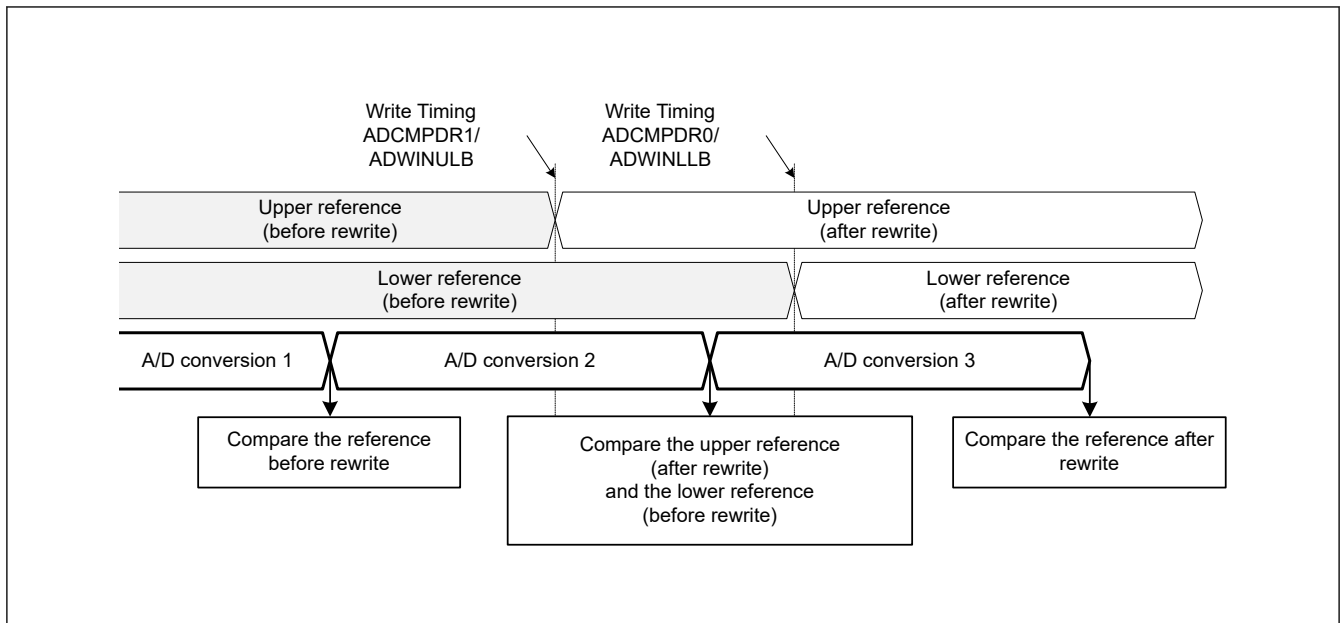
The ADWINULB and ADWINLLB registers specify the reference data when the compare window B function is used. ADWINLLB sets the lower reference for window B, and ADWINULB sets the upper reference for window B.

ADWINnLB are read/write registers.

ADWINnLB are writable even during A/D conversion. The reference data can be dynamically changed by rewriting register values during A/D conversion<sup>\*1</sup>.

Set these registers so that the upper reference is not less than the lower reference ( $ADWINULB \geq ADWINLLB$ ). ADWINULB are not used when the window function is disabled.

Note 1. The lower and the upper references are changed when each register is written. For example, when the upper reference value is changed and the lower reference value is being changed, the MCU compares the upper reference (after rewrite), and the lower reference (before rewrite) with the A/D conversion result. See [Figure 42.6](#). If the comparison during the rewriting of these two references is erroneous, then rewrite these reference values when both ADCSR.ADST and the target Compare Window Operation Enable bit (ADCMPCR.CMPAE or ADCMPCR.CMPBE) are 0.



**Figure 42.6 Comparison between upper and lower references before and after a rewrite**

The ADWINnLB registers use different formats depending on the following conditions:

- The value of A/D Data Register Format Select bit (flush-right or flush-left)
- The value of the A/D Conversion Accuracy Select bit (12-bit, 10-bit, 8-bit)
- The value of A/D-Converted Value Addition/Average Channel Select bits (A/D-converted value addition mode selected or not selected).



The data formats for each condition are shown as follows:

1. When A/D-converted value addition mode is not selected
  - Flush-right data with 12-bit accuracy — Lower 12 bits ([11:0]) are valid
  - Flush-right data with 10-bit accuracy — Lower 10 bits ([9:0]) are valid
  - Flush-right data with 8-bit accuracy — Lower 8 bits ([7:0]) are valid
  - Flush-left data with 12-bit accuracy — Upper 12 bits ([15:4]) are valid
  - Flush-left data with 10-bit accuracy — Upper 10 bits ([15:6]) are valid
  - Flush-left data with 8-bit accuracy — Upper 8 bits ([15:8]) are valid
2. When A/D-converted value addition mode is selected
  - Flush-right data with 12-bit accuracy — Lower 14 bits ([13:0]) are valid
  - Flush-right data with 10-bit accuracy — Lower 12 bits ([11:0]) are valid
  - Flush-right data with 8-bit accuracy — Lower 10 bits ([9:0]) are valid
  - Flush-left data with 12-bit accuracy — Upper 14 bits ([15:2]) are valid
  - Flush-left data with 10-bit accuracy — Upper 12 bits ([15:4]) are valid
  - Flush-left data with 8-bit accuracy — Upper 10 bits ([15:6]) are valid

#### 42.2.28 ADCMPSTR0 : A/D Compare Function Window A Channel Status Register 0

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x0A0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPSTCHA TCHA 15	CMPSTCHA TCHA 14	CMPSTCHA TCHA 13	CMPSTCHA TCHA 12	CMPSTCHA TCHA 11	CMPSTCHA TCHA 10	CMPSTCHA TCHA 09	CMPSTCHA TCHA 08	CMPSTCHA TCHA 07	CMPSTCHA TCHA 06	CMPSTCHA TCHA 05	CMPSTCHA TCHA 04	CMPSTCHA TCHA 03	CMPSTCHA TCHA 02	CMPSTCHA TCHA 01	CMPSTCHA TCHA 00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPSTCHA15 to CMPSTCHA00 <sup>*1</sup>	Compare Window A Flag When Window A operation is enabled (ADCMPCR.CMPAE = 1b), these bits indicate the comparison result of channels to which Window A comparison conditions are applied. Bit 15 (CMPSTCHA15) is associated with ANm15 and bit 0 (CMPSTCHA00) is associated with ANm00. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W

Note: S-TYPE-3, P-TYPE-3

Note: Bits associated with non-existent pins are reserved. This bit is read as 0. The write value should be 0.

Note 1. 00 to 08 (unit 0), 00 to 05 (unit 1) is available.

#### CMPSTCHAN flags (Compare Window A Flag)

The CMPSTCHAN flags indicate the comparison results for channels to which Window A comparison conditions are applied. When a comparison condition set in ADCMPLR0.CMPLCHA is met at the end of A/D conversion, the associated CMPSTCHAN flag sets to 1. When the ADCMPCR.CMPAIE bit is 1, a compare interrupt request (ADC12i\_CMPAI ( $i = 0, 1$ )) is generated when this flag sets to 1.

Writing 1 to the CMPSTCHAN flags is invalid.

[Setting condition]

- The condition set in ADCMPLR0.CMPLCHA is met when ADCMPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

## 42.2.29 ADCMPSER : A/D Compare Function Window A Extended Input Channel Status Register

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x0A4

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CMPS TOCA	CMPS TTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPSTTSA	Compare Window A Temperature Sensor Output Compare Flag When Window A operation is enabled (ADCMPPCR.CMPAE = 1), this bit indicates the temperature sensor output comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
1	CMPSTOCA	Compare Window A Internal Reference Voltage Compare Flag When Window A operation is enabled (ADCMPPCR.CMPAE = 1), this bit indicates the internal reference voltage comparison result. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The ADCMPSER register stores compare results of compare function window A.

### CMPSTTSA flag (Compare Window A Temperature Sensor Output Compare Flag)

The CMPSTTSA flag indicates the temperature sensor output comparison result. When the comparison condition set in ADCMPPLER.CMPLTSA is met at the end of A/D conversion, this flag sets to 1. When the ADCMPPCR.CMPAIE bit is 1, a compare interrupt request (ADC12i\_CMPAI ( $i = 0, 1$ )) is generated when this flag sets to 1.

Writing 1 to the CMPSTTSA flag is invalid.

[Setting condition]

- The condition set in ADCMPPLER.CMPLTSA is met when ADCMPPCR.CMPAE = 1.

[Clearing condition]

- Writing 0 after reading 1.

### CMPSTOCA flag (Compare Window A Internal Reference Voltage Compare Flag)

The CMPSTOCA flag indicates the internal reference voltage comparison result. When the comparison condition set in ADCMPPLER.CMPLOCA is met at the end of A/D conversion, this flag sets to 1. When the ADCMPPCR.CMPAIE bit is 1, a compare interrupt request (ADC12i\_CMPAI ( $i = 0, 1$ )) is generated when this flag sets to 1.

Writing 1 to the CMPSTOCA flag is invalid.

[Setting condition]

- The condition set in ADCMPPLER.CMPLOCA is met when ADCMPPCR.CMPAE = 1.

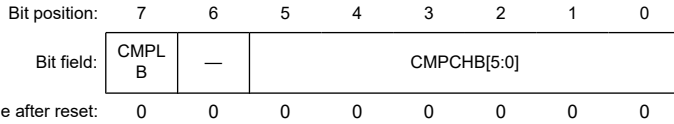
[Clearing condition]

- Writing 0 after reading 1.

### 42.2.30 ADCMPBNSR : A/D Compare Function Window B Channel Select Register

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x0A6



Bit	Symbol	Function	R/W		
5:0	CMPCHB[5:0]	Compare Window B Channel Select These bits select channels to be compared with the compare Window B conditions.	R/W		
		<b>CMPCHB[5:0]</b>		<b>Unit 0</b>	<b>Unit 1</b>
		0x00		AN000	AN100
		0x01		AN001	AN101
		0x02		AN002	AN102
		0x03		AN003 / VBATT 1/3 voltage monitor output	AN103 / VBATT 1/3 voltage monitor output
		0x04		AN004	AN104
		0x05		AN005	AN105
		0x06		AN006	Do not select
		0x07		AN007	Do not select
		0x08		AN008	Do not select
		0x09		Do not select	Do not select
		⋮		⋮	⋮
		0x20		Temperature sensor	
0x21	Internal reference voltage				
0x3F	Do not select				
Others	Setting prohibited				
6	—	This bit is read as 0. The write value should be 0.	R/W		
7	CMPLB	Compare Window B Comparison Condition Setting This bit sets comparison conditions for channels for Window B. The comparison conditions are shown in <a href="#">Figure 42.7</a> .  0: When window function is disabled (ADCMPCR.WCMPE = 0): ADWINLLB value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADWINLLB value, or ADWINULB value < A/D-converted value  1: When window function is disabled (ADCMPCR.WCMPE = 0): ADWINLLB value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADWINLLB value < A/D-converted value < ADWINULB value	R/W		

Note: S-TYPE-3, P-TYPE-3

#### CMPCHB[5:0] bits (Compare Window B Channel Select)

The CMPCHB[5:0] bits specify the channels to be compared with the compare Window B conditions from AN000 to AN002, AN004 to AN008, AN100 to AN102, AN104, AN105, the temperature sensor, the internal reference voltage and the VBATT 1/3 voltage monitor output. The compare Window B function is enabled by specifying the hexadecimal number of the A/D conversion channel selected in the ADANSA0, ADANSB0 registers.

Set the CMPCHB[5:0] bits while the ADCSR.ADST bit is 0.

**CMPLB bit (Compare Window B Comparison Condition Setting)**

The CMPLB bit specifies the comparison conditions for channels for Window B. When the comparison result of an analog input meets the set condition, the associated ADCMPBSR.CMPSTB flag sets to 1 and a compare interrupt request (ADC12i\_CMPBI (i = 0, 1)) is generated.

Compare conditions when the window function is disabled			
CMPLB = 0		CMPLB = 1	
ADWINLLB value $\leq$ A/D converted value	Not met	ADWINLLB value $<$ A/D converted value	Met
ADWINLLB value $>$ A/D converted value	Met	ADWINLLB value $\geq$ A/D converted value	Not met
Compare conditions when the window function is enabled			
CMPLB = 0			
A/D converted value $>$ ADWINULB value		Met	
ADWINLLB value $\leq$ A/D converted value $\leq$ ADWINULB value		Not met	
A/D converted value $<$ ADWINLLB value		Met	
CMPLB = 1			
A/D converted value $\geq$ ADWINULB value		Not met	
ADWINLLB value $<$ A/D converted value $<$ ADWINULB value		Met	
A/D converted value $\leq$ ADWINLLB value		Not met	

**Figure 42.7 Explanation of compare conditions for compare function Window B**

**42.2.31 ADCMPBSR : A/D Compare Function Window B Status Register**

Base address: ADC12m = 0x4033\_2000 + 0x0200 × m (m = 0, 1)  
 ADC12m\_NS = 0x5033\_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x0AC

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CMPS TB

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	CMPSTB	Compare Window B Flag When Window B operation is enabled (ADCMPCR.CMPBE = 1), this bit indicates the comparison result of channels to which Window B comparison conditions are applied, temperature sensor output, internal reference voltage. 0: Comparison conditions are not met. 1: Comparison conditions are met.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

### CMPSTB flag (Compare Window B Flag)

The CMPSTB flag indicates the comparison result of channels to which Window B comparison conditions are applied, the temperature sensor output, internal reference voltage. When the comparison condition set in ADCMPBNSR.CMPLB is met at the end of A/D conversion, this flag sets to 1. When the ADCMPCR.CMPBIE bit is 1, a compare interrupt request (ADC12i\_CMPBI (i = 0, 1)) is generated when this flag sets to 1.

Writing 1 to the CMPSTB flag is invalid.

[Setting condition]

- The condition set in ADCMPBNSR.CMPLB is met when ADCMPCR.CMPBE = 1.

[Clearing condition]

- Writing 0 after reading 1.

### 42.2.32 ADWINMON : A/D Compare Function Window A/B Status Monitor Register

Base address: ADC12m = 0x4033\_2000 + 0x0200 × m (m = 0, 1)  
ADC12m\_NS = 0x5033\_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x08C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MONC MPB	MONC MPA	—	—	—	MONC OMB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MONCOMB	Combination Result Monitor This bit indicates the combination result. This bit is valid when both Window A and Window B operations are enabled. 0: Window A/B composite conditions are not met. 1: Window A/B composite conditions are met.	R
3:1	—	These bits are read as 0.	R
4	MONCMPA	Comparison Result Monitor A 0: Window A comparison conditions are not met. 1: Window A comparison conditions are met.	R
5	MONCMPB	Comparison Result Monitor B 0: Window B comparison conditions are not met. 1: Window B comparison conditions are met.	R
7:6	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

### MONCOMB bit (Combination Result Monitor)

The read-only MONCOMB bit indicates the combined result of comparison condition results A and B based on the combination condition set in the ADCMPCR.CMPAB[1:0] bits.

[Setting condition]

- The combined result meets the combination condition set in the ADCMPCR.CMPAB[1:0] bits when ADCMPCR.CMPAE = 1 and ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The combined result does not meet the combination condition set in the ADCMPCR.CMPAB[1:0] bits.
- ADCMPCR.CMPAE = 0 or ADCMPCR.CMPBE = 0.

#### MONCMPA bit (Comparison Result Monitor A)

The read-only MONCMPA bit is read as 1 when the A/D-converted value of the Window A target channel meets the condition set in ADCMPLR0 and ADCMPLER. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in the ADCMPLR0 and ADCMPLER registers when ADCMPCR.CMPAE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set in the ADCMPLR0 and ADCMPLER registers when ADCMPCR.CMPAE = 1.
- ADCMPCR.CMPAE = 0 (automatically cleared when the ADCMPCR.CMPAE value changes from 1 to 0).

#### MONCMPB bit (Comparison Result Monitor B)

The read-only MONCMPB bit is read as 1 when the A/D-converted value of the Window B target channel meets the condition set in the ADCMPBNSR.CMPLB bit. Otherwise, it is read as 0.

[Setting condition]

- The A/D-converted value meets the condition set in ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1.

[Clearing conditions]

- The A/D-converted value does not meet the condition set in ADCMPBNSR.CMPLB when ADCMPCR.CMPBE = 1.
- ADCMPCR.CMPBE = 0 (automatically cleared when the ADCMPCR.CMPBE value changes from 1 to 0).

### 42.2.33 ADBUFEN : A/D Data Buffer Enable Register

Base address: ADC12m = 0x4033\_2000 + 0x0200 × m (m = 0, 1)  
ADC12m\_NS = 0x5033\_2000 + 0x0200 × m (m = 0, 1)

Offset address: 0x0D0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BUFEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUFEN	Data Buffer Enable 0: The data buffer is not used. 1: The data buffer is used.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The ADBUFEN register sets whether to enable the data buffer.

#### BUFEN bit (Data Buffer Enable)

This bit enables the use of the data buffer.

When BUFEN = 1b, A/D conversion result (addition result) other than self-diagnosis result is stored in ADBUFn.

Disable the data storage operation (BUFEN = 0b) before reading ADBUFPTR.

Do not use the data buffer for data duplexing, or group scan.

### 42.2.34 ADBUFPTR : A/D Data Buffer Pointer Register

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address: 0x0D2

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PTRO VF	BUFPtr[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	BUFPtr[3:0]	Data Buffer Pointer These bits indicate the number of data buffer to which the next A/D converted data is transferred.	R/W
4	PTROVF	Pointer Overflow Flag 0: The data buffer pointer has not overflowed. 1: The data buffer pointer has overflowed.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

ADBUFPtr is a register that indicates the data buffer pointer and overflow status.

#### BUFPtr[3:0] bit (Data Buffer Pointer)

These bits indicate the number of data buffer to which the next A/D converted data is transferred.

When data has been transferred to data buffer 15, the pointer value becomes 0000b and the PTROVF bit is set to 1.

When the next data has been transferred, the data in data buffer 0 is overwritten.

Writing 0x00 to this register clears the value of these bits. Writing a value other than 0x00 is disabled.

#### PTROVF bit (Pointer Overflow Flag)

This bit indicates whether the data buffer pointer has overflowed. This bit is set to 1 when the pointer value becomes 0000b (overflow).

Writing 0x00 to this register clears this bit value. Writing a value other than 0x00 is disabled.

### 42.2.35 ADBUFn : A/D Data Buffer Registers n (n = 0 to 15)

Base address:  $ADC12m = 0x4033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )  
 $ADC12m\_NS = 0x5033\_2000 + 0x0200 \times m$  ( $m = 0, 1$ )

Offset address:  $0x0B0 + 0x2 \times n$  ( $n = 0$  to 15)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADBUF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADBUF[15:0]	Converted Value 15 to 0 Functions vary depending on the selected mode and accuracy. See <a href="#">Table 42.24</a> and <a href="#">Table 42.25</a> .	R

Note: S-TYPE-3, P-TYPE-3

ADBUFn registers are 16-bit read-only registers that sequentially store all A/D conversion results. The automatic clear function is not applied to these registers.

ADBUFn settings are the same as the A/D data register format settings.

The following conditions determine the formats for data in the ADBUFn registers:

- Setting of the Register Format Select bit (ADCER.ADRFMT) (flush-left or flush-right)
- Setting of the Addition/Average Count Select bits (ADADC.ADC[2:0]) (1, 2, 3, 4, or 16 times)
- Setting of the Average Mode Enable bit (ADADC.AVEE) (addition or average).

This section describes the data formats for these conditions in different modes.

### (1) When A/D-converted value addition/average mode is not selected

Table 42.24 shows the bit assignment for each accuracy.

**Table 42.24 Bit assignment for each accuracy**

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	These bits are read as 0.				Converted Value 11 to 0: 12-bit A/D-converted value											
Left-justified data with 12-bit accuracy	Converted Value 11 to 0: 12-bit A/D-converted value												These bits are read as 0.			

### (2) When A/D-converted value average mode is selected

A/D-converted value average mode can be selected when 2 or 4 times is specified in the A/D-converted value addition mode. When A/D converted value average mode is selected, These registers indicate the mean of A/D-converted values on a specific channel. The value is stored in the A/D data register based on the setting of the A/D Data Register Format Select bit in the same way as for normal A/D conversion.

### (3) When A/D-converted value addition mode is selected

For 12-bit, 10-bit, 8-bit accuracy, 1, 2, 3, or 4 times can be selected in the A/D-converted value addition mode. A/D conversion results are stored in the A/D data register as a 2-bit-extended value of the specified conversion accuracy.

For 12-bit accuracy, 16 times can also be selected in the A/D-converted value addition mode. In A/D-converted value addition mode, these registers indicate the value that is obtained by adding A/D-converted values on a specific channel. A/D conversion results are stored in the A/D data register as a 4-bit-extended value of the specified conversion accuracy.

When A/D-converted value addition mode is selected, the value is stored in the A/D data register based on the settings of the A/D Data Register Format Select bits.

Table 42.25 shows the bit assignment for each accuracy.

**Table 42.25 Bit assignment for each accuracy when A/D-converted value addition mode is selected**

Accuracy	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Right-justified data with 12-bit accuracy	When 16 conversion times is specified		Added Value 15 to 0: 16-bit sum of A/D conversion results													
	When 1, 2, 3, or 4 conversion times is specified		These bits are read as 0.		Added Value 13 to 0: 14-bit sum of A/D conversion results											
Left-justified data with 12-bit accuracy	When 1, 2, 3, or 4 conversion times is specified		Added Value 15 to 0: 16-bit sum of A/D conversion results													
	When 16 conversion times is specified		Added Value 13 to 0: 14-bit sum of A/D conversion results													These bits are read as 0.

## 42.3 Operation

### 42.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

Scan conversion is performed in any of the three operating modes:



- Single scan mode
- Continuous scan mode
- Group scan mode

In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until software sets the ADCSR.ADST bit to 0. In group scan mode, the selected channels in group A, B are scanned once after scan starts in response to the respective synchronous trigger.

In single scan mode and continuous scan mode, A/D conversion is performed for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for the ANn channels in group A selected in the ADANSA0 register, and for the ANn channels in group B selected in the ADANSB0 register, starting from the channel with the smallest number n.

When self-diagnosis is selected, it is executed once at the beginning of each scan and one of the three reference voltages is converted.

The temperature sensor output and internal reference voltage can be selected at the same time as the analog input of the channels, and A/D conversion is performed on the analog input of channels, temperature sensor output, and internal reference voltage, in that order.

Double trigger mode can be used with single scan mode or group scan mode. With double trigger mode enabled (ADCSR.DBLE = 1), A/D conversion data of a channel selected in the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by the synchronous trigger (GPT, ELC) selected in the ADSTRGR.TRSA[5:0] bits. In group scan mode, only group A can use double trigger mode.

In the extended operation of double trigger mode, the A/D conversion operation is generated from the synchronous trigger combination selected in the ADSTRGR.TRSA[5:0] bits. In addition to normal double trigger mode operation, A/D conversion data with odd number trigger (ELC\_AD00 (unit 0), ELC\_AD10 (unit 1), GTCIADAn (Unit0) and GTCIADAm (Unit1) (n = 0 to 3, m = 4, 5)) is stored in A/D Data Duplexing Register A (ADDBLDRA), and A/D conversion data with even number trigger (ELC\_AD01 (unit 0), ELC\_AD11 (unit 1), GTCIADBn (Unit0) and GTCIADBm (Unit1) (n = 0 to 3, m = 4, 5)) is stored in A/D Data Duplexing Register B (ADDBLDRB). In the extended operation of double trigger mode, when one of the trigger combinations occurs at the same time, the data duplexing register settings for the specified triggers do not work, and A/D conversion data is stored in A/D Data Duplexing Register B (ADDBLDRB).

The ADC12 ignores a synchronous trigger that occurs during the A/D conversion started by another synchronous trigger.

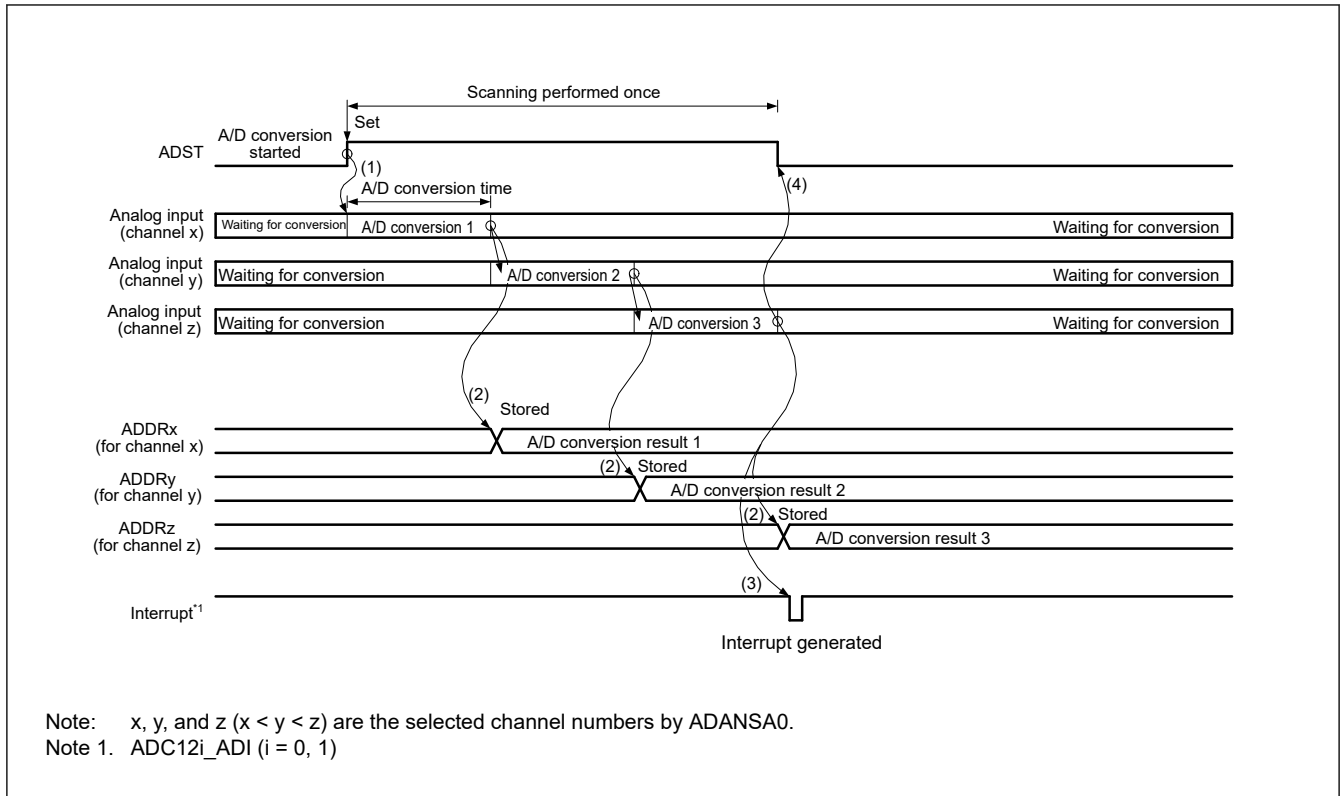
When any of the channels AN000, AN001 (unit 0) are set as a channel-dedicated sample-and-hold circuit in the SHANS[1:0] bits in ADSHCR, the target analog input specified is sampled and held before the first A/D conversion of each scan.

## 42.3.2 Single Scan Mode

### 42.3.2.1 Basic Operation (Without Channel-Dedicated Sample-and-Hold Circuits)

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (GPT, ELC), or an asynchronous trigger input, A/D conversion is performed for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
2. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the associated A/D data register (ADDRy).
3. When A/D conversion of all the selected channels is completed, an ADC12i\_ADI (i = 0, 1) interrupt request is generated.
4. The ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion of all the selected channels is completed. The ADC12 then enters a wait state.



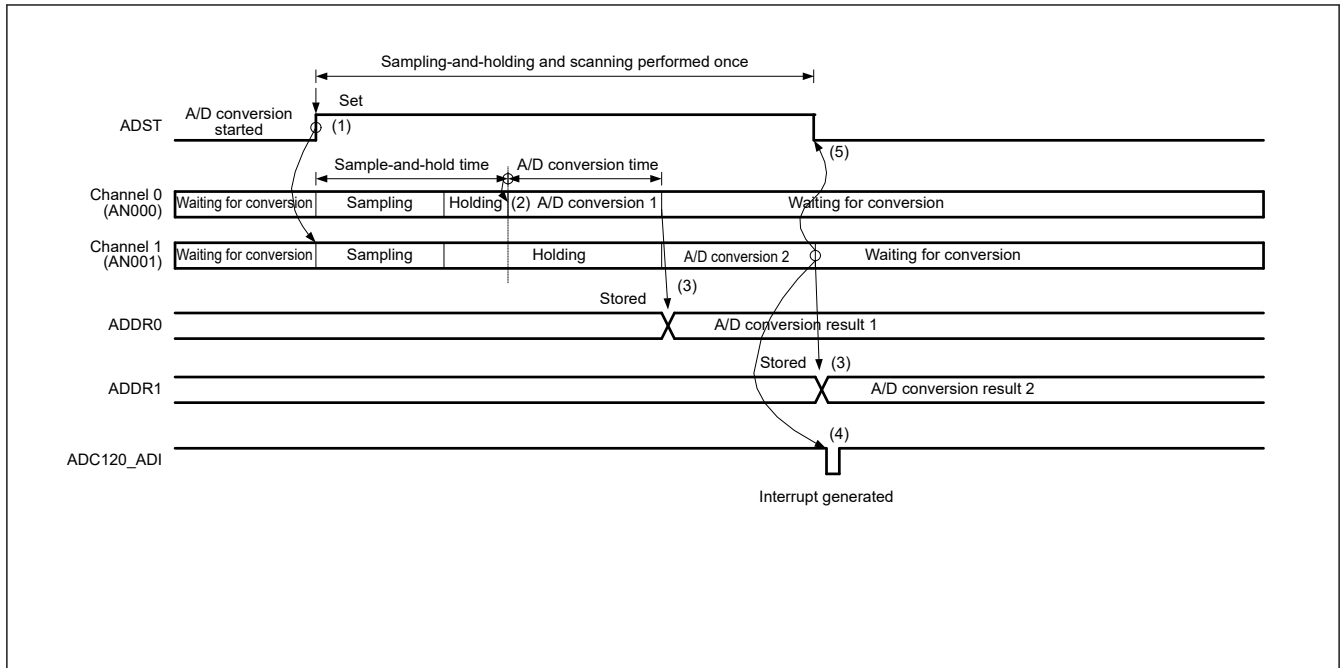
**Figure 42.8 Example basic operation in single scan mode when the analog inputs (channel x to z) are selected**

### 42.3.2.2 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Disabled)

When the channel-dedicated sample-and-hold circuit is used, sample-and-hold operation is first performed, and then A/D conversion is performed on the analog input of all the specified channels once. The channels whose dedicated sample-and-hold circuit is to be used can be selected in the SHANS[1:0] bits in ADSHCR.

The operation is as follows:

1. Analog input sampling of all channels whose dedicated sample-and-hold circuit is to be used starts when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (GPT, ELC), or asynchronous trigger input.
2. After the sample-and-hold operation, A/D conversion is performed on the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC12i\_ADI (i = 0) interrupt request is generated (no register setting).
5. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then, the ADC12 enters a wait state.



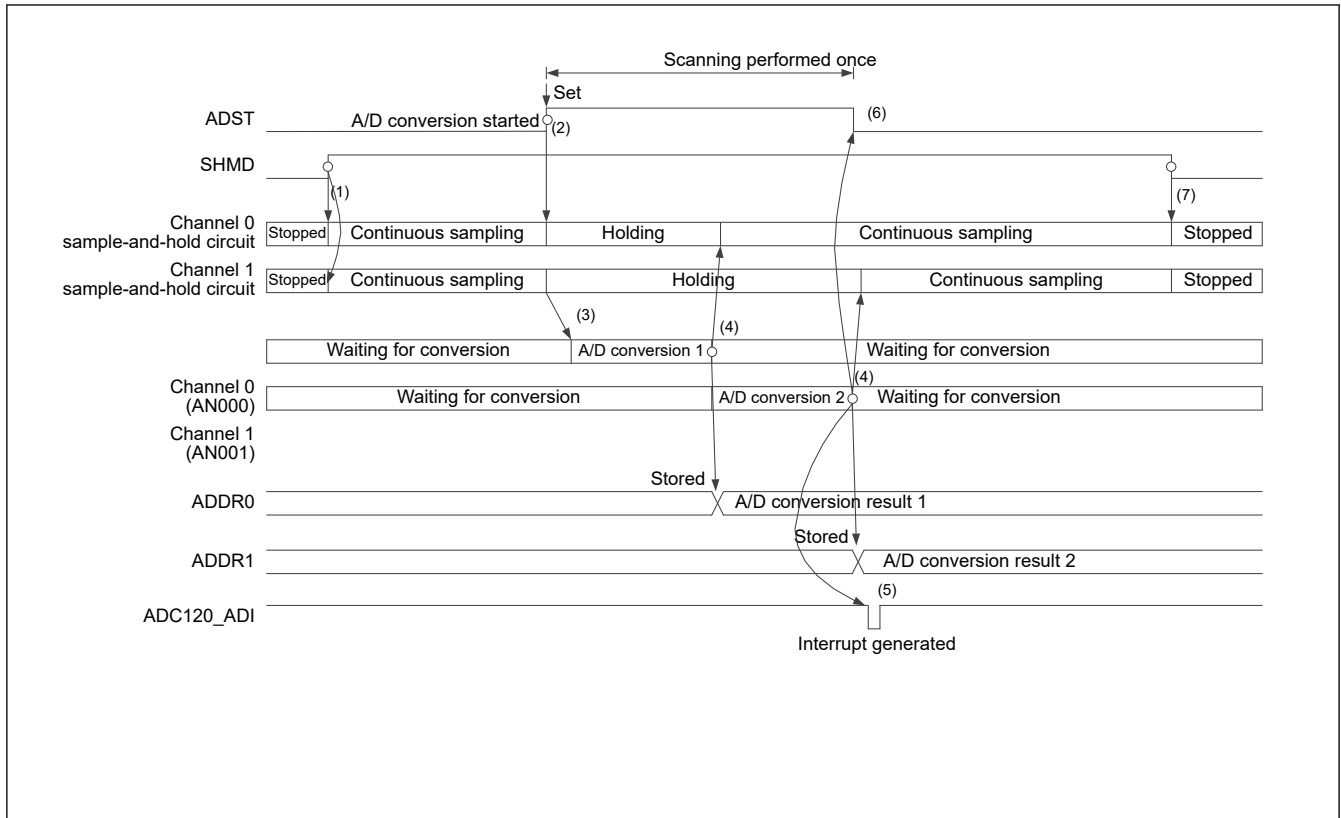
**Figure 42.9** Example operation in single scan mode when the channel-dedicated sample-and-hold circuits are used and AN000, AN001 are selected

#### 42.3.2.3 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Enabled)

When a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operations are performed first, followed by A/D conversion on the analog inputs on all selected channels, once. The ADSHCR.SHANS[1:0] bits specify the channels for which the channel-dedicated sample-and-hold circuits are to be used.

The operation is as follows:

1. When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[1:0] bits start continuous sampling.
2. Analog input holding starts for all the channels for which the channel-dedicated sample-and-hold circuits are to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (GPT, ELC), or input of an asynchronous trigger.
3. After the stabilization time of the sample-and-hold circuits elapses, A/D conversion is performed on the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
5. When A/D conversion of all the selected channels completes, an ADC12i\_ADI (i = 0) interrupt request is generated (no register setting).
6. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then, the ADC12 enters a wait state. If this is followed by single scanning, set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).
7. When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits stop.

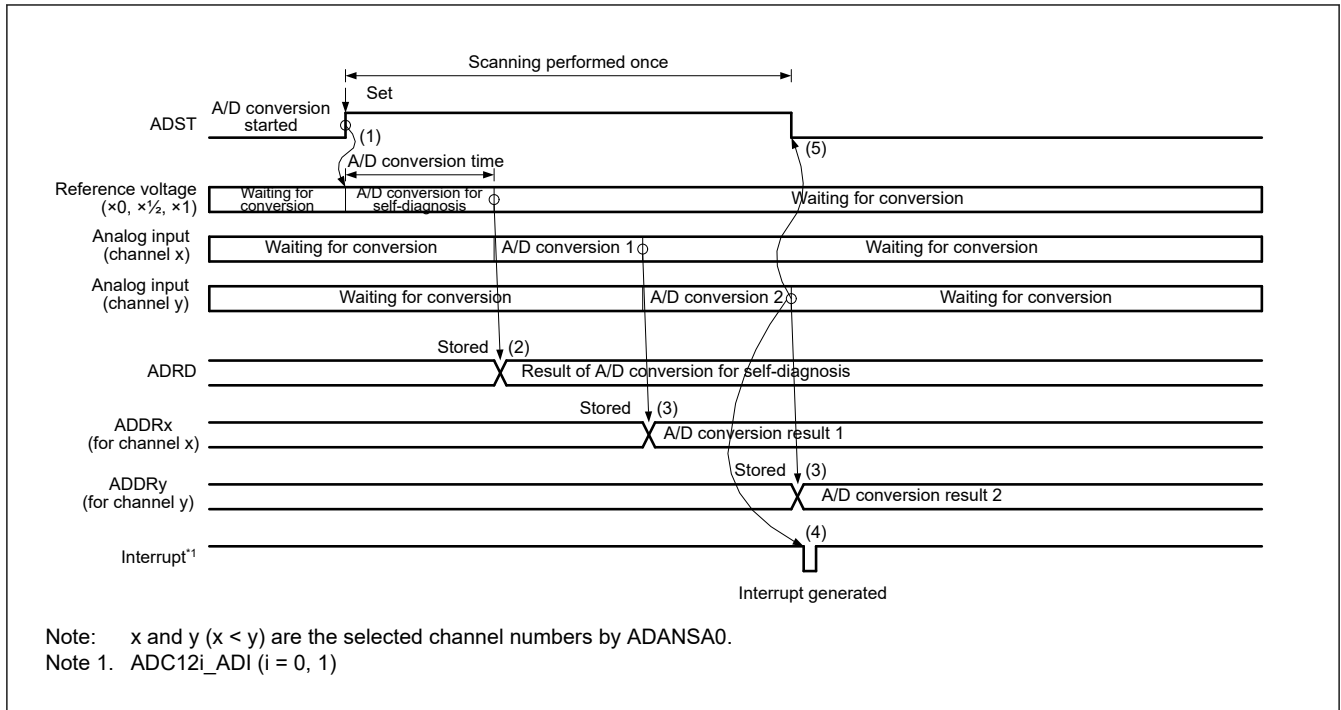


**Figure 42.10** Example operation in single scan mode when channel-dedicated sample-and-hold circuits are used, AN000, AN001 are selected, and continuous sampling is enabled

#### 42.3.2.4 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits)

When channels and self-diagnosis are selected, A/D conversion is first performed for the reference voltage VREFH0, VREFH ( $\times 0$ ,  $\times 1/2$ , or  $\times 1$ ), then A/D conversion is performed once on the analog input of the selected channels as follows:

1. A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger input, a synchronous trigger input (GPT, ELC), or an asynchronous trigger input.
2. When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADDRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the associated A/D data register (ADDRy).
4. When A/D conversion of all the selected channels is completed, an ADC12i\_ADI ( $i = 0, 1$ ) interrupt request is generated.
5. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion of all the selected channels is completed. The ADC12 then enters a wait state.



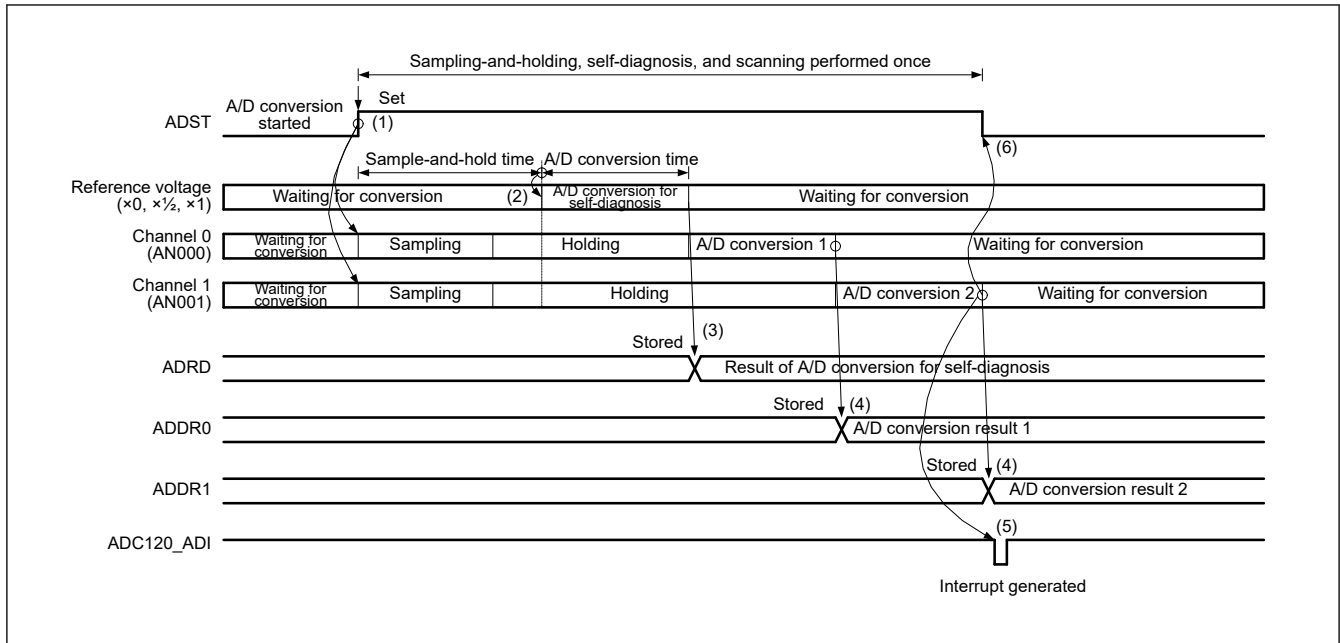
**Figure 42.11 Example basic operation in single scan mode when the analog inputs (channel x and y) are selected with self-diagnosis**

#### 42.3.2.5 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Disabled)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is disabled, the sample-and-hold operation is performed first, and then A/D conversion is performed once for the reference voltage VREFH0 (unit 0) ( $\times 0$ ,  $\times 1/2$ , or  $\times 1$ ) supplied to the ADC12. After that, A/D conversion is performed only once on the analog input of the selected channels.

The operation is as follows:

1. Analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (GPT, ELC), or asynchronous trigger input.
2. After the sample-and-hold operation, A/D conversion for self-diagnosis starts.
3. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADDRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
5. When A/D conversion of all the selected channels completes, an ADC12i\_ADI (i = 0) interrupt request is generated (no register setting).
6. The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then, the ADC12 enters a wait state.



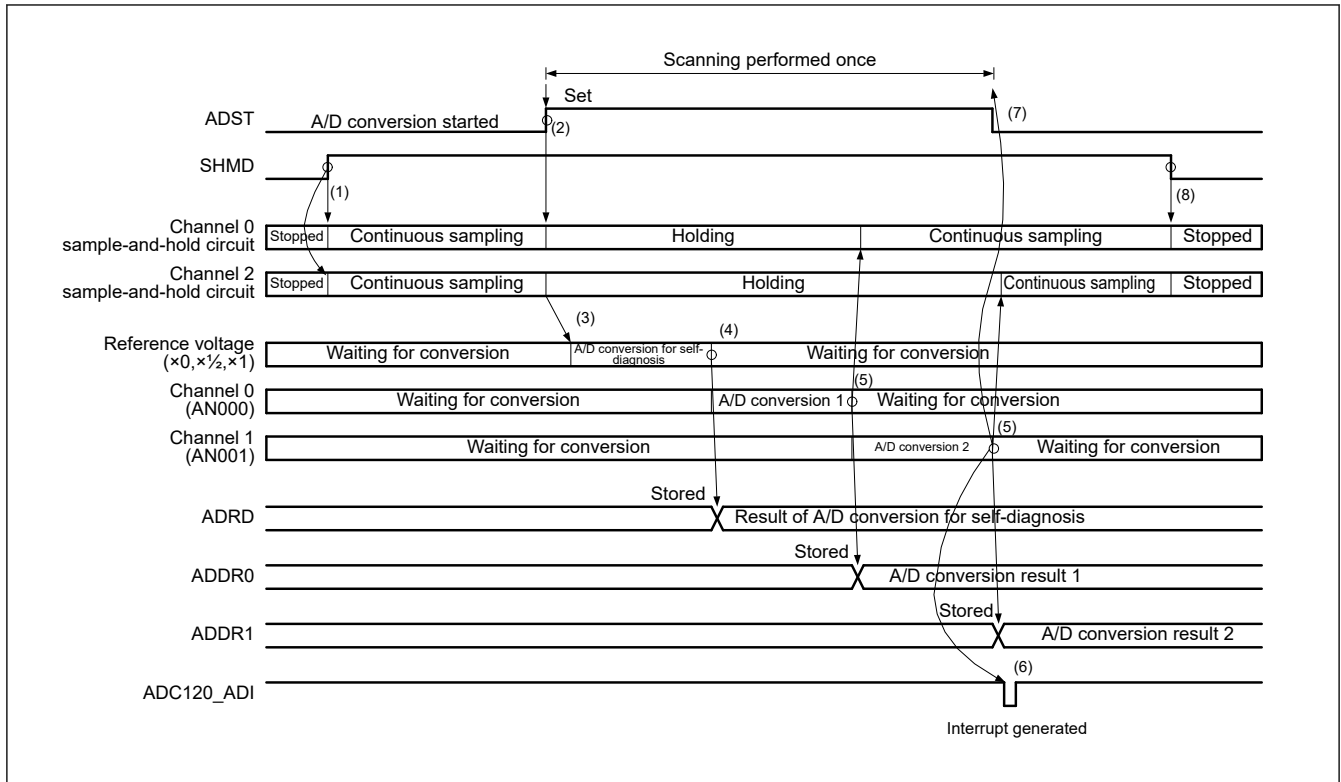
**Figure 42.12 Example operation in single scan mode when channel-dedicated sample-and-hold circuits are used, AN000, AN001 are selected with self-diagnosis, and continuous sampling is disabled**

### 42.3.2.6 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Enabled)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operations are performed first, followed by A/D conversion of the reference voltage VREFH0 (unit 0) supplied to the ADC12. After that, A/D conversion is performed only once on the analog input of the selected channels.

The operation is as follows:

1. When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[1:0] bits start continuous sampling.
2. Analog input holding starts for all the channels for which the channel-dedicated sample-and-hold circuits are to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (GPT, ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 kΩ) elapse after the ADSHMSR.SHMD bit is set to 1.
3. After the stabilization time of the sample-and-hold circuits elapse, A/D conversion for self-diagnosis starts.
4. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
5. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
6. When A/D conversion of all the selected channels completes, an ADC12i\_ADI (i = 0) interrupt request is generated (no register setting).
7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels completes. Then, the ADC12 enters a wait state. If this is followed by single scanning, set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).
8. When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits stop.



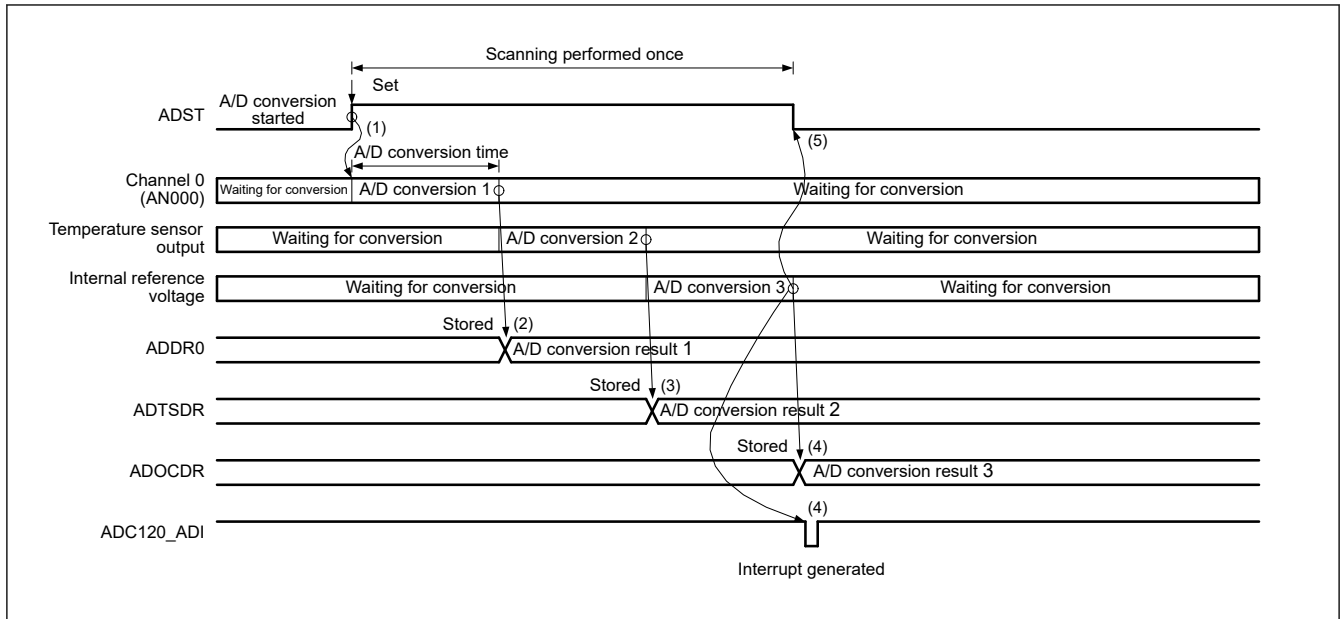
**Figure 42.13 Example operation in single scan mode when channel-dedicated sample-and-hold circuits are used, AN000, AN001 are selected with self-diagnosis, and continuous sampling is enabled**

### 42.3.2.7 A/D Conversion of Temperature Sensor Output or Internal Reference Voltage

When the channels and temperature sensor output or internal reference voltage are selected at the same time, A/D conversion is performed first on the analog input of the selected channels, and once on the temperature sensor output or internal reference voltage. When both temperature sensor output and internal reference voltage are selected, A/D conversion of the temperature sensor output and internal reference voltage is performed, in that order. With the channels deselected, selecting only the temperature sensor output or internal reference voltage is also possible.

The operation is as follows:

1. When a software trigger, synchronous trigger (GPT, ELC), or asynchronous trigger sets the ADCSR.ADST bit to 1 (A/D conversion start), A/D conversion is performed for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
2. On completion of A/D conversion on the channels, the result is stored in the associated A/D Data Register y (ADDRy), and then A/D conversion of the temperature sensor output starts.
3. On completion of A/D conversion of the temperature sensor output, the result is stored in the associated A/D Temperature Sensor Data Register (ADTSDR), and then A/D conversion of the internal reference voltage starts.
4. On completion of A/D conversion of the internal reference voltage, the result is stored in the associated A/D Internal Reference Voltage Data Register (ADOCDR), and an ADC12i\_ADI (i = 0, 1) interrupt request is generated (no register setting).
5. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 on completion of A/D conversion. Then, the ADC12 enters a wait state.



**Figure 42.14 Example basic operation in single scan mode when AN000 and temperature sensor output or internal reference voltage are selected**

#### 42.3.2.8 A/D Conversion in Double-Trigger Mode

When double trigger mode is selected in single scan mode, two rounds of single scan operation started by a synchronous trigger (GPT, ELC) are performed in sequence.

Deselect self-diagnosis and set the temperature sensor output A/D conversion select bit (ADEXICR.TSSA and ADEXICR.TSSB) and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA and ADEXICR.OCSB) to 0.

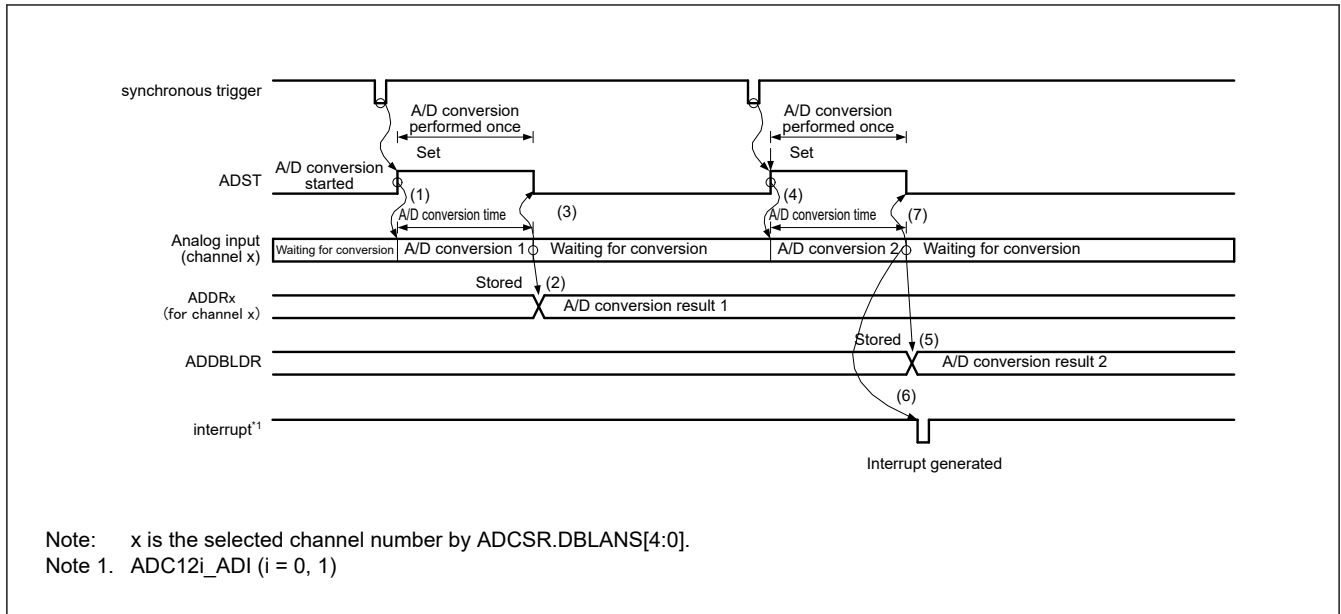
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated in the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 register is invalid.

In double trigger mode, select a synchronous trigger (GPT, ELC) with the ADSTRGR.TRSA[5:0] bits. Additionally, set the ADCSR.EXTRG bit to 0 and the ADCSR.TRGE bit to 1. Do not use a software trigger.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a synchronous trigger input (GPT, ELC), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
2. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
3. The ADCSR.ADST bit is automatically set to 0 and the ADC12 enters a wait state. An ADC12i\_ADI (i = 0, 1) interrupt request is not generated.
4. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input, A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
5. When A/D conversion is completed, the result is stored in the A/D Data Duplexing Register (ADDBLDR), which is exclusively used in double-trigger mode.
6. An ADC12i\_ADI (i = 0, 1) interrupt request is generated.
7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion is completed. Then the ADC12 enters a wait state.





**Figure 42.15 Example operation in single scan mode when double-trigger mode is selected and the analog input (channel x) is duplicated**

#### 42.3.2.9 Extended Operations when Double-Trigger Mode is Selected

When double trigger mode is selected in single scan mode, and a synchronous trigger (ELC\_AD00 (unit 0) and ELC\_AD10 (unit 1)/ELC\_AD01 (unit 0) and ELC\_AD11 (unit 1), GTCIADA0 (unit 0) and GTCIADA4 (unit 1) / GTCIADB0 (Unit0) and GTCIADB4 (unit 1), GTCIADA1 (unit 0) and GTCIADA5 (unit 1) / GTCIADB1 (Unit0) and GTCIADB5 (unit 1), GTCIADA2 (Unit 0) / GTCIADB2 (Unit0) or GTCIADA3 (Unit 0) / GTCIADB3 (Unit0)) is selected as the trigger for the start of A/D conversion, two rounds of single scan operation are performed.

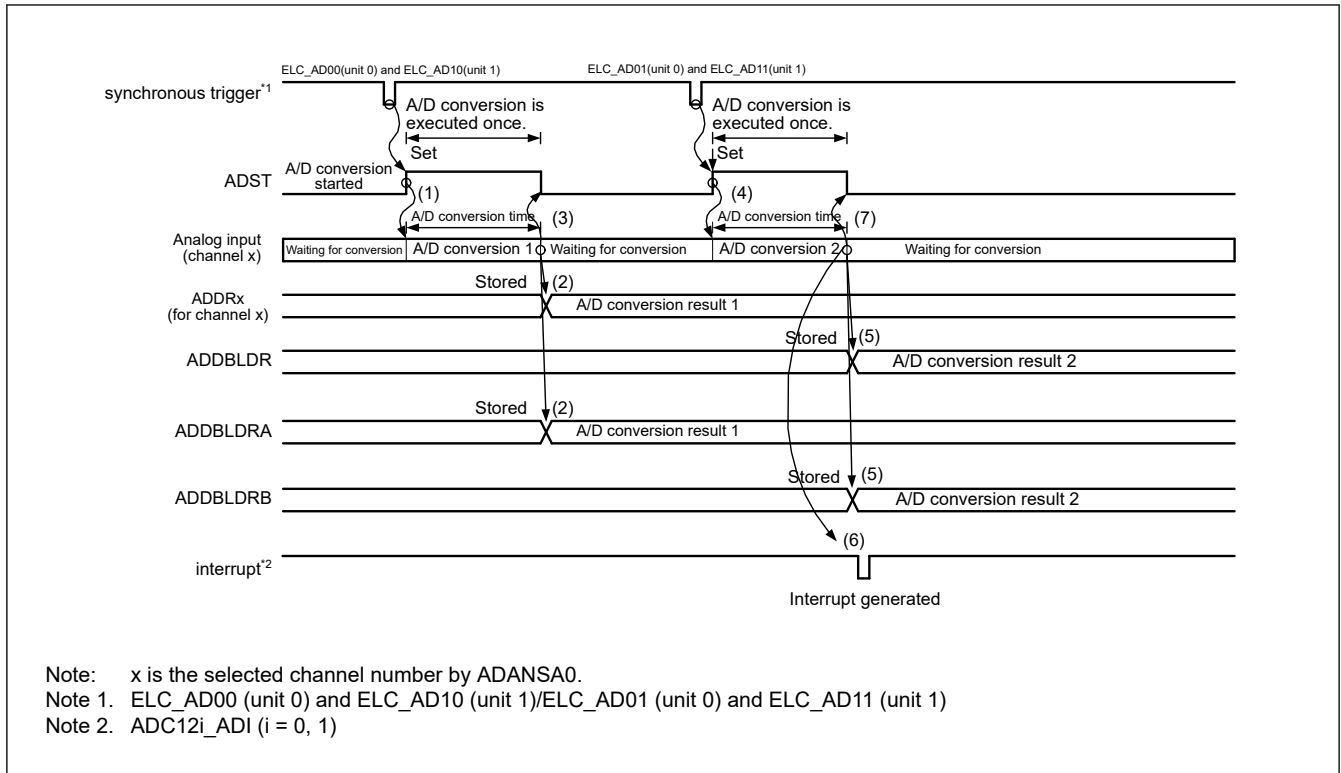
Deselect self-diagnosis and set the temperature sensor output A/D conversion select bit (ADEXICR.TSSA and ADEXICR.TSSB), and the internal reference voltage A/D conversion select bit (ADEXICR.OCSA and ADEXICR.OCSB) to 0.

Duplication of A/D conversion data is enabled by setting the channel number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 register is invalid.

In extended double trigger mode, select a synchronous trigger combination by setting the ADSTRGR.TRSA[5:0] bits to 0x0B, set the ADCSR.EXTRG bit to 0, and set the ADCSR.TRGE bit to 1. Do not use a software trigger.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a synchronous trigger input, A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
2. When A/D conversion completes, the A/D conversion result is stored in the associated A/D Data Register (ADDRy) and in A/D Data Duplexing Register A (ADDBLDRA) or A/D Data Duplexing Register B (ADDBLDRB) when the ELC\_ADi0 or ELC\_ADi1 trigger is input respectively (i = 0, 1).
3. The ADCSR.ADST bit is automatically set to 0 and the ADC12 enters a wait state. An ADC12i\_ADI (i = 0, 1) interrupt request is not generated.
4. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger (ELC\_AD00 (unit 0) and ELC\_AD10 (unit 1)/ELC\_AD01 (unit 0) and ELC\_AD11 (unit 1)), A/D conversion starts on the single channel selected in the ADCSR.DBLANS[4:0] bits.
5. When A/D conversion completes, the A/D conversion result is stored in the A/D Data Duplexing Register (ADDBLDR) and in A/D Data Duplexing Register A (ADDBLDRA) or A/D Data Duplexing Register B (ADDBLDRB) when the ELC\_ADi0 or ELC\_ADi1 trigger is input respectively (i = 0, 1).
6. An ADC12i\_ADI (i = 0, 1) interrupt request is generated.
7. The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion and is automatically set to 0 when A/D conversion completes. The ADC12 then enters a wait state.



**Figure 42.16 Example extended operation in double trigger mode with duplication selected for the analog input (channel x) and ELC\_AD00 (unit 0) and ELC\_AD10 (unit 1)/ELC\_AD01 (unit 0) and ELC\_AD11 (unit 1)**

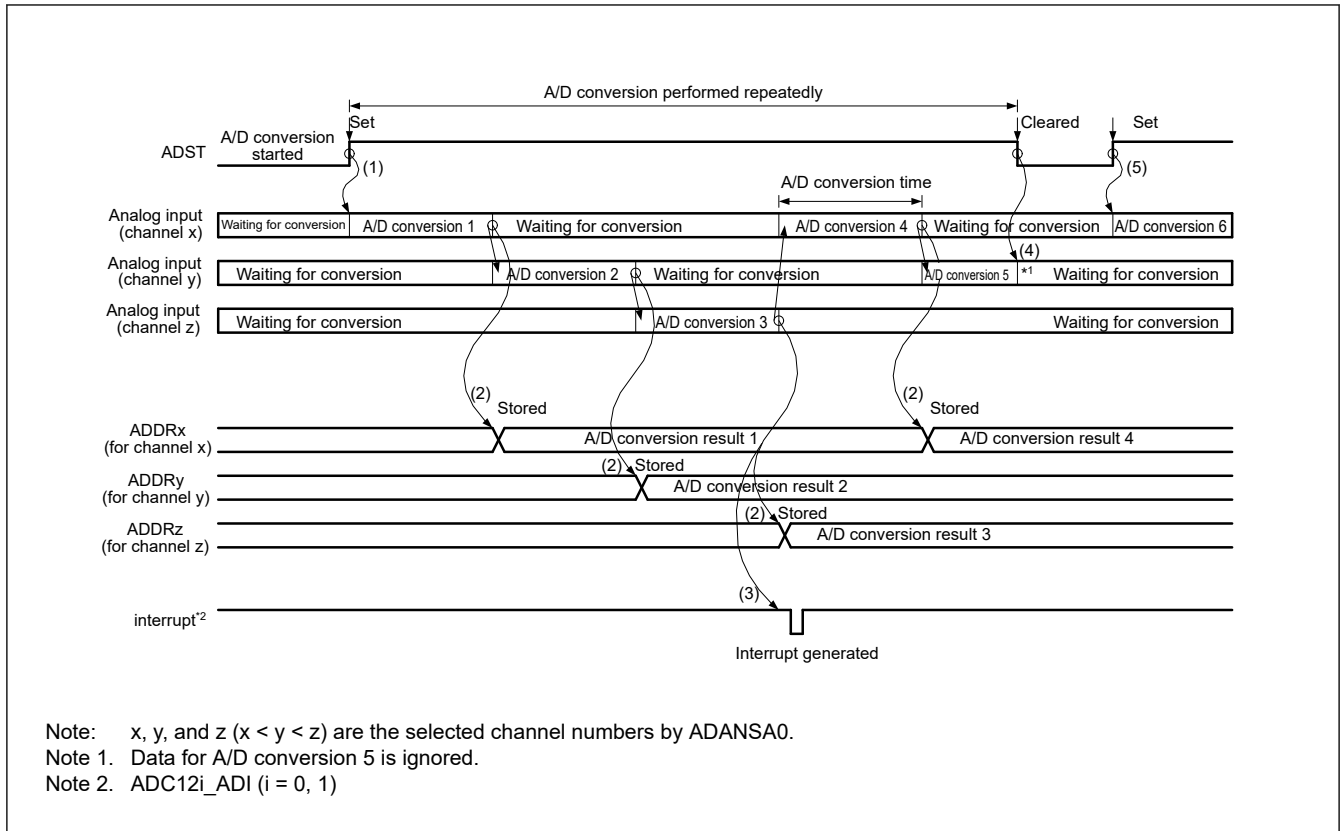
### 42.3.3 Continuous Scan Mode

#### 42.3.3.1 Basic Operation (Without Channel-Dedicated Sample-and-Hold Circuits)

In continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, a synchronous trigger input (GPT, ELC), or an asynchronous trigger input, A/D conversion is performed for ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
2. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the associated A/D Data Register (ADDRy).
3. When A/D conversion of all the selected channels is completed, an ADC12i\_ADI (i = 0, 1) interrupt request is generated. The ADC12 sequentially starts A/D conversion for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
4. The ADCSR.ADST bit is not automatically cleared, and steps 2. and 3. are repeated as long as ADCSR.ADST remains 1 (A/D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
5. When the ADCSR.ADST bit is later set to 1 (A/D conversion start), A/D conversion starts again for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.



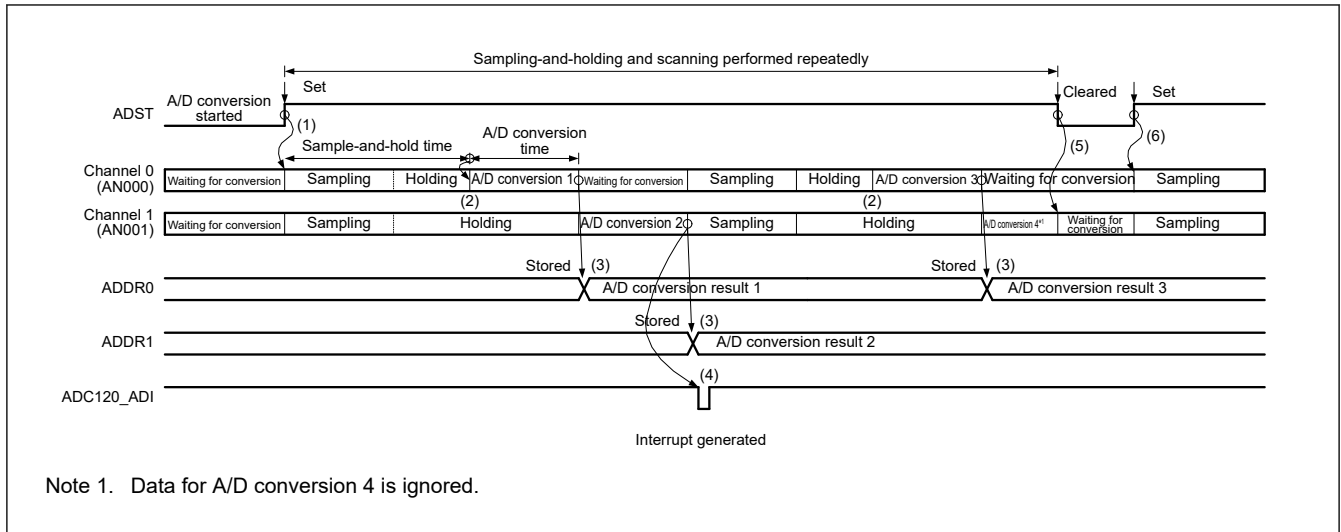
**Figure 42.17 Example basic operation in continuous scan mode when the analog inputs (channel x to z) are selected**

### 42.3.3.2 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Disabled)

When the channel-dedicated sample-and-hold circuit is used with the continuous sampling disabled, sample-and-hold operation is performed first, and then A/D conversion is repeated on the analog input of all the specified channels. The channels whose dedicated sample-and-hold circuit is to be used can be selected in the SHANS[1:0] bits in ADSHCR.

The operation is as follows:

1. Analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (GPT, ELC), or asynchronous trigger input.
2. After the sample-and-hold operation, A/D conversion is performed for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy).
4. When A/D conversion of all the selected channels completes, an ADC12i\_ADI (i = 0) interrupt request is generated (no register setting). At the same time, analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used.
5. The ADST bit is not automatically cleared, and steps 2 to 4 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
6. When the ADST bit is then set to 1 (A/D conversion start), analog input sampling starts again for all the channels whose dedicated sample-and-hold circuit is to be used.



**Figure 42.18** Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used and AN000 and AN001 are selected

#### 42.3.3.3 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Enabled)

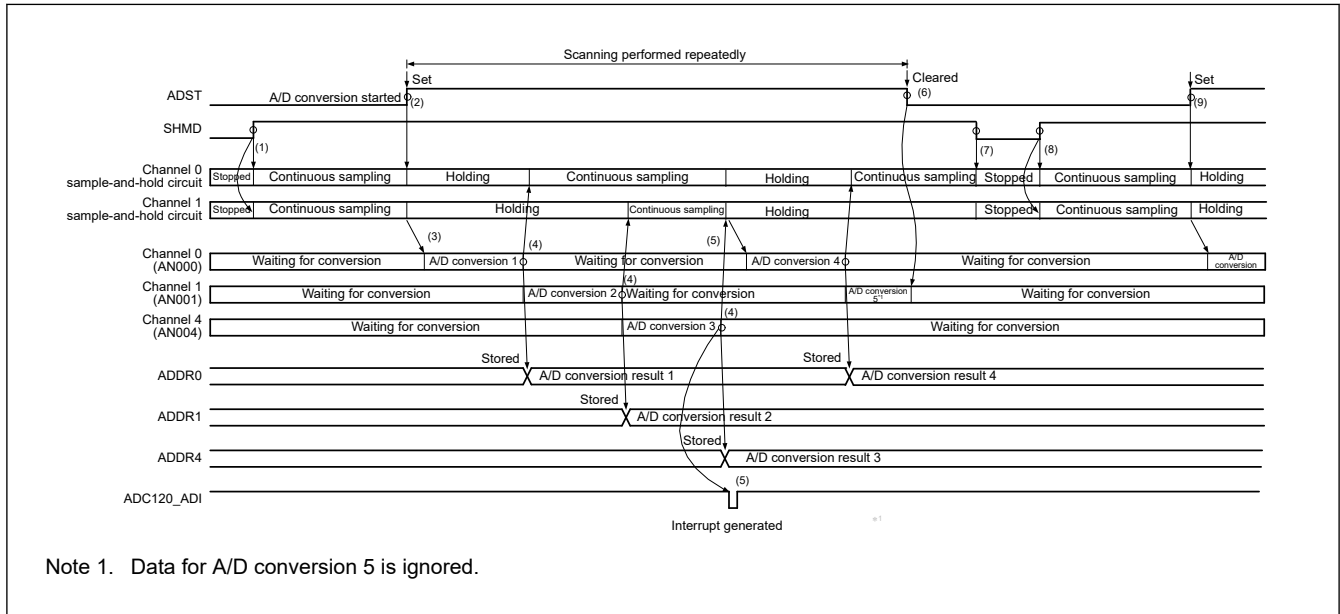
When a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, sample-and-hold operations are performed first, after which the analog inputs on all the selected channels are A/D-converted as described in this section. The channels for which the channel-dedicated sample-and-hold circuits are to be used can be selected in the ADSHCR.SHANS[1:0] bits.

The operation is as follows:

1. When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[1:0] bits start continuous sampling.
2. Analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (GPT, ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 k $\Omega$ ) elapse after the ADSHMSR.SHMD bit is set to 1.
3. After the stabilization time of the sample-and-hold circuits elapses, A/D conversion is performed for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
5. When A/D conversion of all the selected channels completes, an ADC12i\_ADI (i = 0) interrupt request is generated (no register setting). Also, analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used.
6. The ADCSR.ADST bit is not automatically cleared, and steps 3 to 5 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
7. When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits stop.
8. When the ADSHMSR.SHMD bit is then set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[1:0] bits start continuous sampling.
9. When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used.

**Note:** If continuous scanning is performed when only those channels with the sample-and-hold circuits are selected, time for continuous sampling cannot be secured in the second and subsequent continuous scans. When continuous sampling by the channel-dedicated sample-and-hold circuits is enabled for continuous scanning, select one or more channels among AN002 to AN008, temperature sensor output, internal reference voltage and VBATT 1/3 voltage monitor output for unit 0, and AN100 to AN105, temperature sensor output, internal reference voltage and VBATT

1/3 voltage monitor output for unit 1, and set the continuous sampling time for the sample-and-hold circuits to at least 400 ns (when the permissible signal source impedance is 1 kΩ).



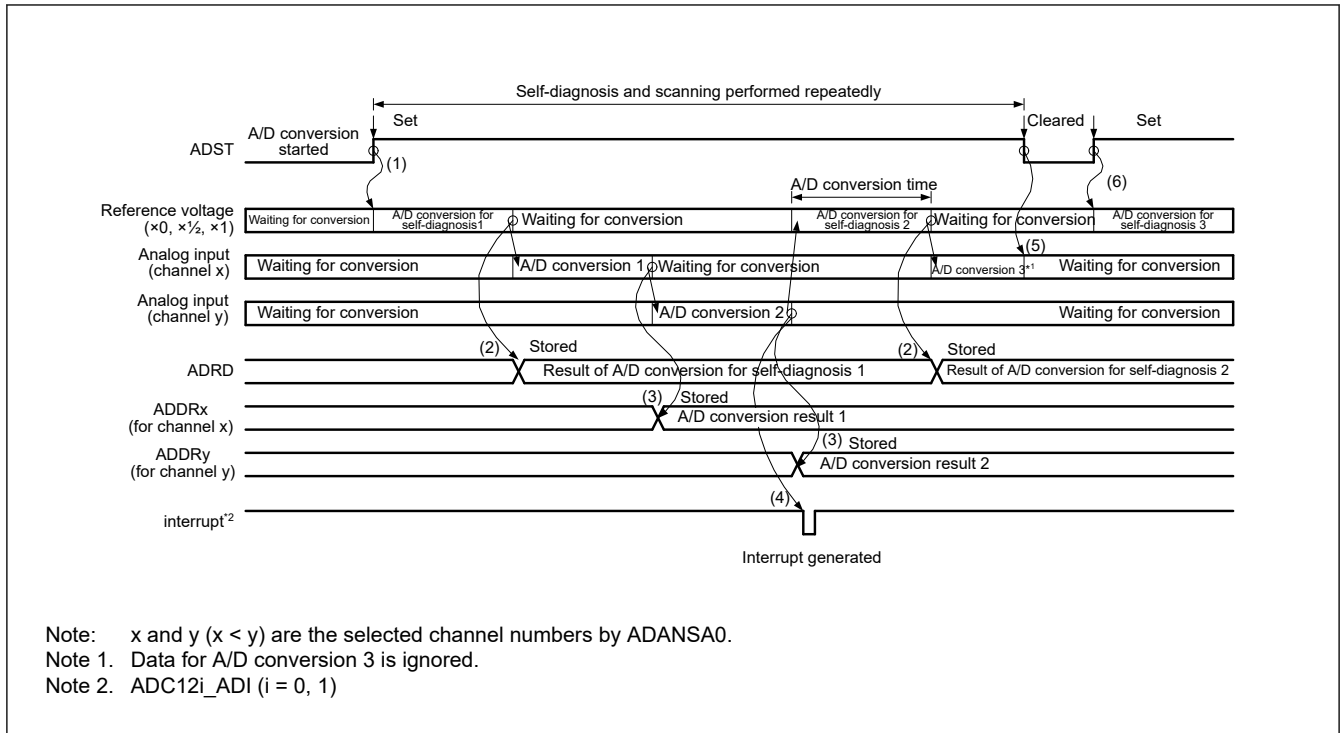
**Figure 42.19 Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used, AN000, AN001, and AN004 are selected, and continuous sampling is enabled**

#### 42.3.3.4 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits)

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage VREFH0, VREFH ( $\times 0$ ,  $\times 1/2$ , or  $\times 1$ ) supplied to the ADC12, and A/D conversion is performed on the analog input of the selected channels. This sequence is repeated as described in the section that follows.

The operation is as follows:

1. A/D conversion for self-diagnosis is first started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger input, a synchronous trigger input (GPT, ELC), or an asynchronous trigger input.
2. When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADDRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
3. Each time A/D conversion of a single channel is completed, the A/D conversion result is stored in the corresponding A/D Data Register (ADDRy).
4. When A/D conversion of all the selected channels is completed, an ADC12i\_ADI ( $i = 0, 1$ ) interrupt request is generated. At the same time, the ADC12 starts A/D conversion for self-diagnosis and then on the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
5. The ADCSR.ADST bit is not automatically cleared, and steps 2. to 4. are repeated as long as the ADCSR.ADST bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
6. When the ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.



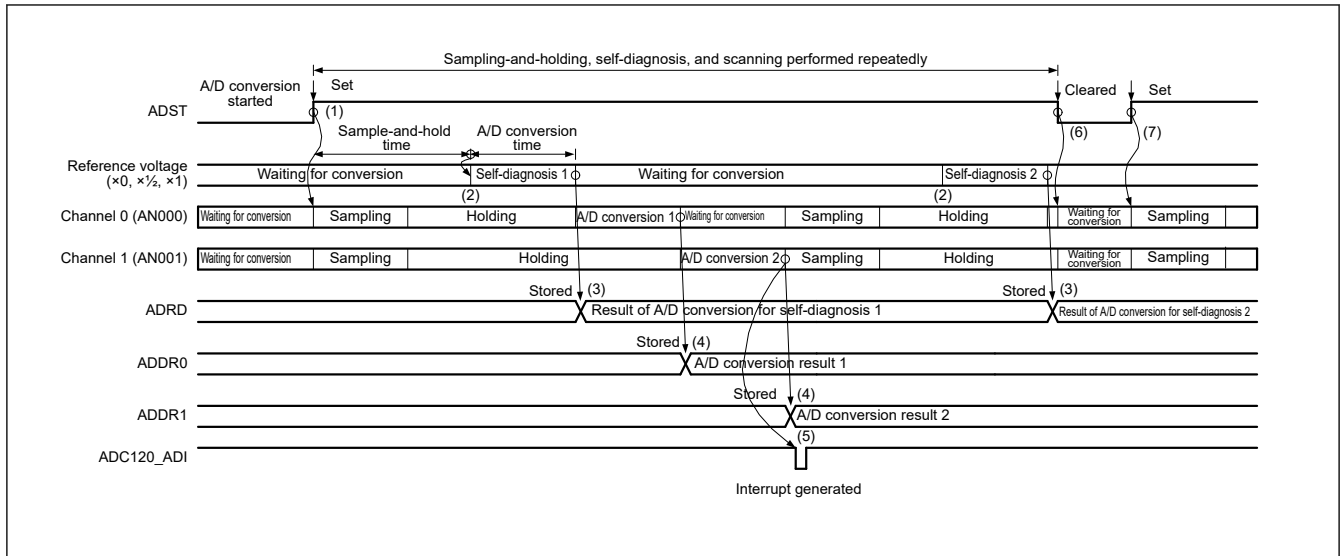
**Figure 42.20 Example basic operation in continuous scan mode when the analog inputs (channel x and y) are selected with self-diagnosis**

### 42.3.3.5 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Disabled)

When the channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is disabled, sample-and-hold operation is performed first, and then A/D conversion is performed for the reference voltage VREFH0 (unit 0) (×0, ×1/2, or ×1) supplied to the ADC12, and then A/D conversion is performed on the analog input of the selected channels, and this sequence is repeated.

The operation is as follows:

1. Analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, synchronous trigger input (GPT, ELC), or asynchronous trigger input.
2. After the sample-and-hold operation, A/D conversion for self-diagnosis starts.
3. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
4. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D data register (ADDRy).
5. When A/D conversion of all the selected channels completes, an ADC12i\_ADI (i = 0) interrupt request is generated (no register setting). At the same time, analog input sampling starts for all channels whose dedicated sample-and-hold circuit is to be used.
6. The ADST bit is not automatically cleared, and steps 2 to 5 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
7. When the ADST bit is then set to 1 (A/D conversion start), analog input sampling starts again for all the channels whose dedicated sample-and-hold circuits are to be used.



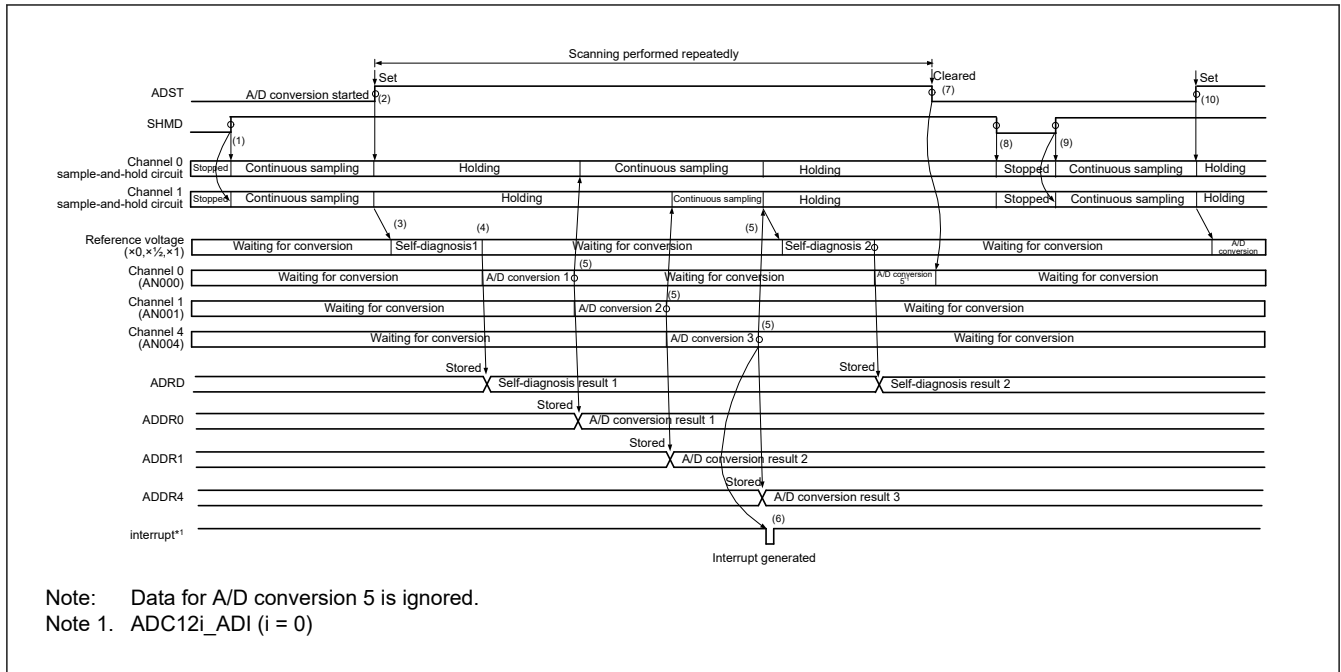
**Figure 42.21 Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used and AN000 and AN001 are selected with self-diagnosis**

#### 42.3.3.6 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits and Continuous Sampling Enabled)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used while continuous sampling is enabled, the sample-and-hold operation is performed first, and followed by A/D conversion of the reference voltage VREFH0 (unit 0) ( $\times 0$ ,  $\times 1/2$ , or  $\times 1$ ) supplied to the ADC12. After that, A/D conversion is performed on the analog input of the selected channels, and this sequence is repeated.

The operation is as follows:

1. When the ADSHMSR.SHMD bit is set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[1:0] bits start continuous sampling.
2. Analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used when the ADCSR.ADST bit is set to 1 (A/D conversion start) by a software trigger, input of a synchronous trigger signal (GPT, ELC), or input of an asynchronous trigger. Set the ADCSR.ADST bit to 1 after at least 400 ns (when the permissible signal source impedance is 1 k $\Omega$ ) elapse after the ADSHMSR.SHMD bit is set to 1.
3. After the stabilization time of the sample-and-hold circuits elapses, A/D conversion for self-diagnosis starts.
4. When A/D conversion for self-diagnosis completes, the A/D conversion result is stored in the A/D Self-Diagnosis Data Register (ADRD). A/D conversion is then performed for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
5. Each time A/D conversion of a single channel completes, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy), and the sample-and-hold circuit restarts continuous sampling.
6. When A/D conversion of all the selected channels completes, an ADC12i\_ADI (i = 0) interrupt request is generated (no register setting). Also, analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used.
7. The ADCSR.ADST bit is not automatically cleared, and steps 3 to 6 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
8. When the ADSHMSR.SHMD bit is set to 0, the sample-and-hold circuits stop.
9. When the ADSHMSR.SHMD bit is then set to 1, the sample-and-hold circuits selected in the ADSHCR.SHANS[1:0] bits start continuous sampling.
10. When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input holding starts for all channels for which the channel-dedicated sample-and-hold circuits are to be used.



**Figure 42.22 Example operation in continuous scan mode when channel-dedicated sample-and-hold circuits are used, AN000, AN001, and AN004 are selected with self-diagnosis, and continuous sampling is enabled**

#### 42.3.3.7 A/D Conversion of Temperature Sensor Output or Internal Reference Voltage

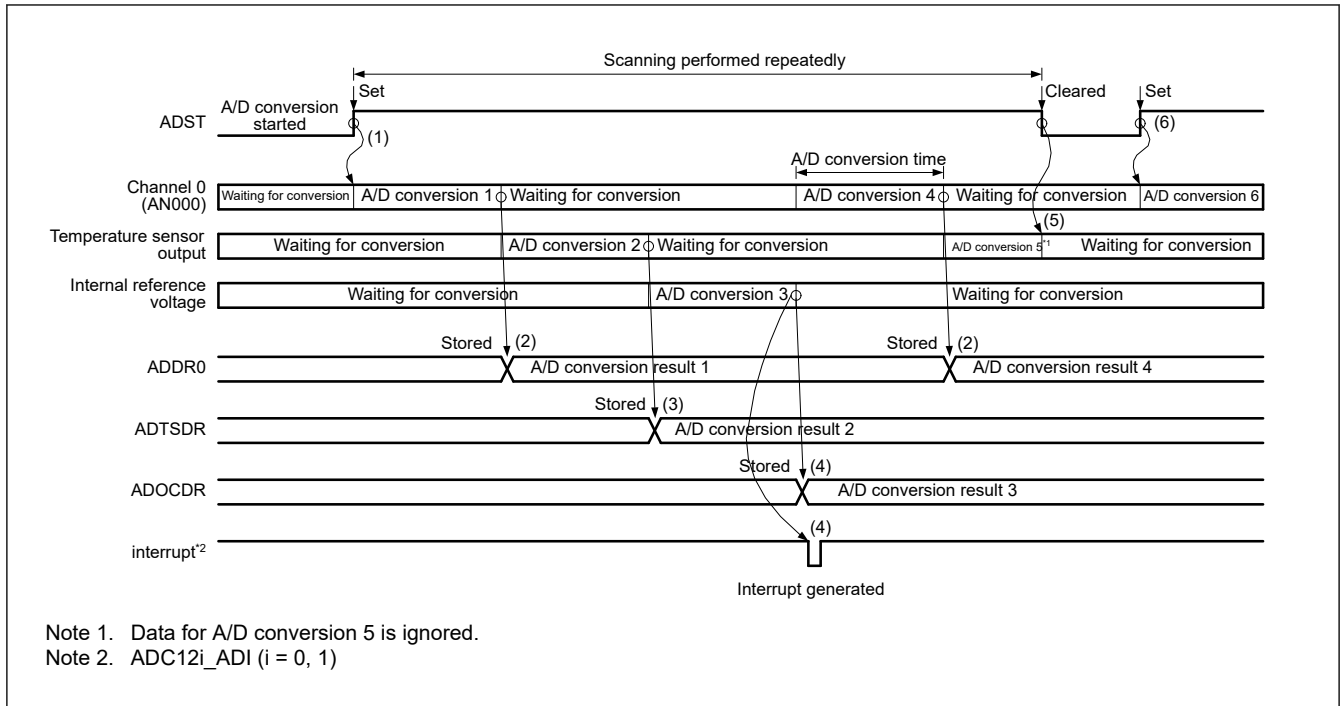
When the channels and temperature sensor output or internal reference voltage are selected at the same time, A/D conversion is first performed on the analog input of the selected channels, and then the A/D conversion of the temperature sensor output or internal reference voltage is repeated. When both the temperature sensor output and internal reference voltage are selected, A/D conversion of the temperature sensor output and internal reference voltage is performed, in that order.

With the channels deselected, selecting only the temperature sensor output or internal reference voltage is also possible.

The operation is as follows:

1. When a software trigger, synchronous trigger (GPT, ELC), or asynchronous trigger sets the ADCSR.ADST bit to 1 (A/D conversion start), A/D conversion is performed for the ANn channels selected in the ADANSA0 register, starting from the channel with the smallest number n.
2. On completion of A/D conversion on the channels, the result is stored in the associated A/D Data Register y (ADDRy), and then A/D conversion of temperature sensor output starts.
3. On completion of A/D conversion of the temperature sensor output, the result is stored in the associated A/D Temperature Sensor Data Register (ADTSDR), and then A/D conversion of the internal reference voltage starts.
4. On completion of A/D conversion of the internal reference voltage, the result is stored in the associated A/D Internal Reference Voltage Data Register (ADOCDR), and an ADC12i\_ADI (i = 0, 1) interrupt request is generated. In addition, the ADC12 continuously starts A/D conversion for the ANn channels selected in the ADANSA0 register, starting from the channel with the lowest number n.
5. The ADCSR.ADST bit is not cleared automatically, and steps 2 to 4 are repeated as long as this bit remains set to 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the ADC12 enters a wait state.
6. When the ADCSR.ADST bit is then set to 1 (A/D conversion start), A/D conversion starts again for the ANn channels selected in the ADANSA0 register, starting from the channel with the lowest number n.





**Figure 42.23 Example basic operation in continuous scan mode when AN000 and temperature sensor output or internal reference voltage are selected**

### 42.3.4 Group Scan Mode

#### 42.3.4.1 Basic Operation

In group scan mode, A/D conversion is performed once on the analog input of all the specified channels in group A and B after scanning is started by a synchronous trigger (GPT, ELC). The scan operation of each group is similar to the scan operation in single scan mode.

The synchronous triggers can be selected in the ADSTRGR.TRSA[5:0] bits for group A and in the ADSTRGR.TRSB[5:0] bits for group B. Use different triggers for group A and B to prevent simultaneous A/D conversion of the two groups. Do not use a software trigger.

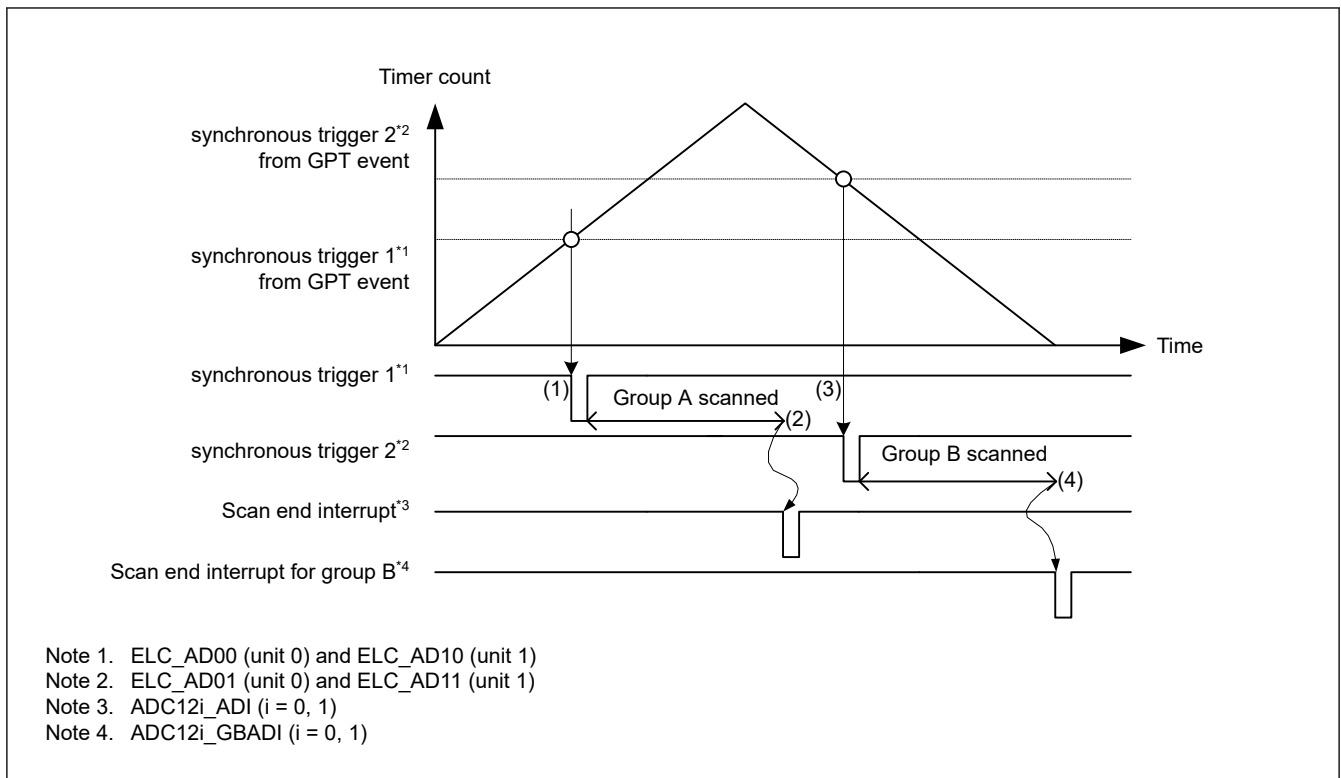
The group A channels to be A/D-converted are selected using the ADANSA0 register and the ADEXICR.TSSA and OCSA bits. The group B channels to be A/D-converted are selected using the ADANSB0 register and the ADEXICR.TSSB and OCSB bits. Group A and B cannot use the same channels.

When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for Group A and B.

The following sequence describes operation in group scan mode using a synchronous trigger from the ELC. In this example, the ELC\_AD00 (unit 0) and ELC\_AD10 (unit 1) trigger from the ELC is used to start conversion of group A and the ELC\_AD01 (unit 0) and ELC\_AD11 (unit 1) trigger from the ELC is used to start conversion of group B. In addition, ELC\_AD00 (unit 0) and ELC\_AD10 (unit 1) and ELC\_AD01 (unit 0) and ELC\_AD11 (unit 1) are selected for the GPT event in the associated ELC.ELSRn registers.

The operation is as follows:

1. Scanning of group A is started by ELC\_AD00 (unit 0) and ELC\_AD10 (unit 1).
2. When group A scanning completes, an ADC12i\_ADI (i = 0, 1) interrupt is generated (no register setting).
3. Scanning of group B is started by ELC\_AD01 (unit 0) and ELC\_AD11 (unit 1).
4. When group B scanning completes, an ADC12i\_GBADI (i = 0, 1) interrupt is generated if the ADCSR.GBADIE bit is 1 (ADC12i\_GBADI (i = 0, 1) interrupt when scanning completion is enabled).



**Figure 42.24 Example basic operation in group scan mode when synchronous triggers from the ELC are used**

#### 42.3.4.2 A/D Conversion in Double-Trigger Mode

When double trigger mode is selected in group scan mode, two rounds of single scan operation started by a synchronous trigger (GPT, ELC) are performed as a sequence for group A. For group B, single scan operation started by a synchronous trigger (GPT, ELC) is performed once.

In group scan mode, the synchronous trigger can be selected in the ADSTRGR.TRSA[5:0] bits for group A and in the ADSTRGR.TRSB[5:0] bits for group B. Use different triggers for group A, B to prevent simultaneous A/D conversion of the two groups. Do not use a software trigger or an asynchronous trigger.

When an ELC\_AD00 (unit 0) and ELC\_AD10 (unit 1)/ELC\_AD01 (unit 0) and ELC\_AD11 (unit 1) is selected as group A synchronous triggers by setting the ADSTRGR.TRSA[5:0] bits to 0x0B, operation proceeds in extended double trigger mode.

The group A channel to be A/D-converted is selected using the DBLANS[4:0] bits in the ADCSR register, while the group B channels to be A/D-converted are selected using the ADANSB0 register. Group A, B cannot use the same channels.

When double-trigger mode is selected in group scan mode, set the A/D conversion select bits for both the temperature sensor output (ADEXICR.TSSA) and the internal reference voltage (ADEXICR.OCSA) to 0 (deselected).

Self-diagnosis cannot be selected when double trigger mode is selected in group scan mode.

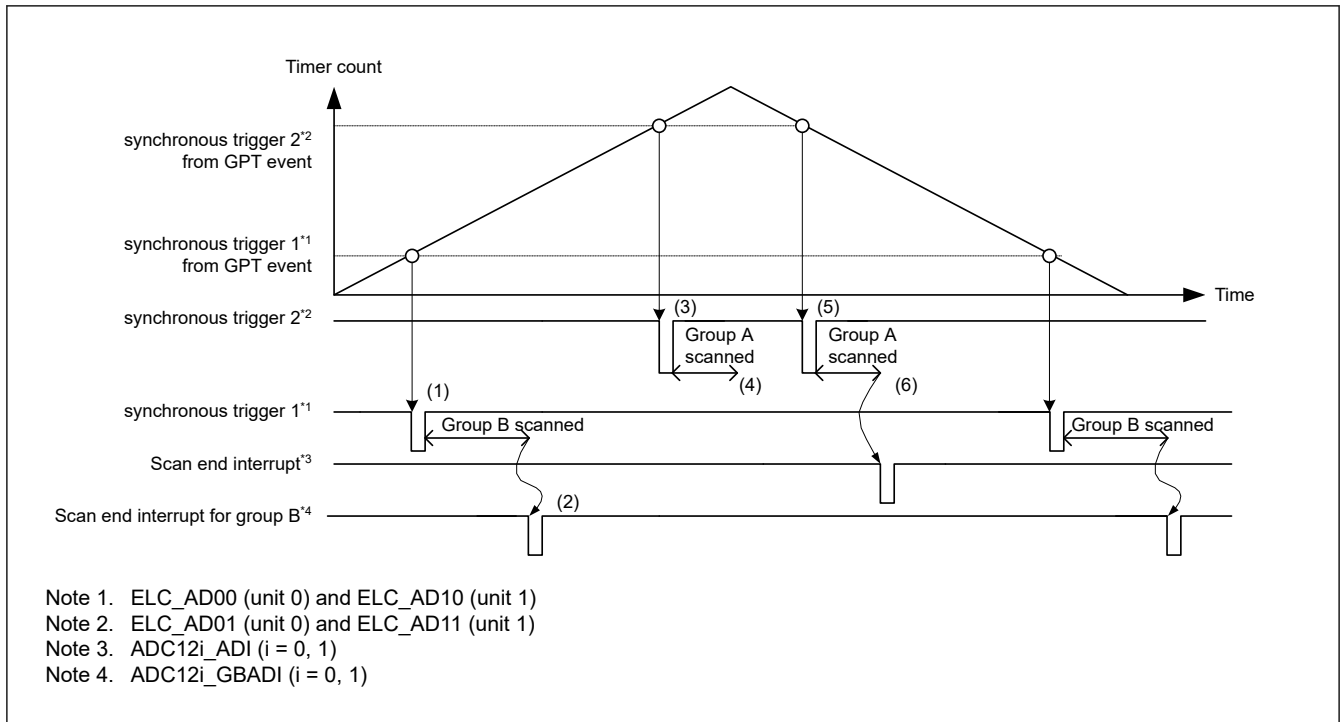
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated in the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1.

The following sequence describes operation in group scan mode with double trigger mode selected and using a synchronous trigger from the ELC. In this example, the ELC\_AD00 (unit 0) and ELC\_AD10 (unit 1) trigger is used to start conversion of group A and the ELC\_AD01 (unit 0) and ELC\_AD11 (unit 1) trigger is used to start conversion of group B. In addition, ELC\_AD00 (unit 0) and ELC\_AD10 (unit 1) and ELC\_AD01 (unit 0) and ELC\_AD11 (unit 1) are selected for the GPT event in the associated ELC.ELSRn registers.

The operation is as follows:

1. Scanning of group B is started by the ELC\_AD00 (unit 0) and ELC\_AD10 (unit 1) trigger from the ELC.
2. When group B scanning completes, an ADC12i\_GBADI (i = 0, 1) interrupt is generated if the GBADIE bit in ADCSR is 1 (ADC12i\_GBADI (i = 0, 1) interrupt when scanning completion is enabled).

3. The first scan of group A is started by the first ELC\_AD01 (unit 0) and ELC\_AD11 (unit 1) trigger.
4. When the first scan of group A completes, the conversion result is stored in the associated A/D Data Register y (ADDRy). An ADC12i\_ADI (i = 0, 1) interrupt request is not generated.
5. The second scan of group A is started by the second ELC\_AD01 (unit 0) and ELC\_AD11 (unit 1) trigger.
6. When the second scan of group A completes, the conversion result is stored in ADDBLDR. An ADC12i\_ADI (i = 0, 1) interrupt is generated.



**Figure 42.25 Example basic operation in group scan mode with double-trigger mode when synchronous triggers from the ELC are used**

#### 42.3.4.3 Group Priority Operation

Group priority operation is performed by setting the ADGSPCR.PGS bit to 1 in group-scan mode. The priority of groups is group A > group B.

When setting the PGS bit in the ADGSPCR register to 1, follow the procedure described in [Figure 42.26](#). If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.

As the basic operation in group-scan mode, a trigger input generated during A/D conversion of group A, B is ignored, and the A/D conversion operation of each group is similar to the operation in single-scan mode.

In group priority operation, if a trigger for a priority group is input during scanning of a lower-priority group, A/D conversion for the lower-priority group is stopped and A/D conversion for the priority group is performed.

If the setting of the ADGSPCR.GBRSCN bit is 0, the lower-priority group enters a wait state when A/D conversion for the priority group completes. A trigger input of the lower-priority group generated during A/D conversion is ignored.

If the setting of the ADGSPCR.GBRSCN bit is 1, A/D conversion for the lower-priority group automatically restarts upon completion of A/D conversion for the priority group. A trigger input of the lower-priority group generated during A/D conversion on the priority group takes effect, and A/D conversion for the lower-priority group is automatically performed upon completion of A/D conversion on the priority group.

If the ADGSPCR.GBRSCN bit is 1 and the ADGSPCR.LGRRS bit is 0, A/D conversion for the lower-priority group is restarted from the first channel. If the setting of the ADGSPCR.LGRRS bit is 1, A/D conversion for the lower-priority group is restarted from the channel for which the conversion stopped. However, if the self-diagnosis function is used, the A/D conversion is restarted from the channel for which the conversion stopped after self-diagnosis completed.

[Table 42.26](#) summarizes operations in response to the input of a trigger during A/D conversion with the settings of the ADGSPCR.GBRSCN bit.

If the setting of the ADGSPCR.GBRP bit is 1, A/D conversion operation for the lowest-priority group is to continuously perform single scans.

For the trigger settings in group-scan mode, select a synchronous trigger for group A by using the ADSTRGR.TRSA[5:0] bits, a synchronous trigger for group B by using the ADSTRGR.TRSB[5:0] bits. Each trigger must be different from each other. Set the ADSTRGR.TRSB[5:0] bits to 0x3F when setting the ADGSPCR.GBRP bit to 1.

The channels to be scanned must be selected in the registers shown in [section 42.3.4. Group Scan Mode](#).

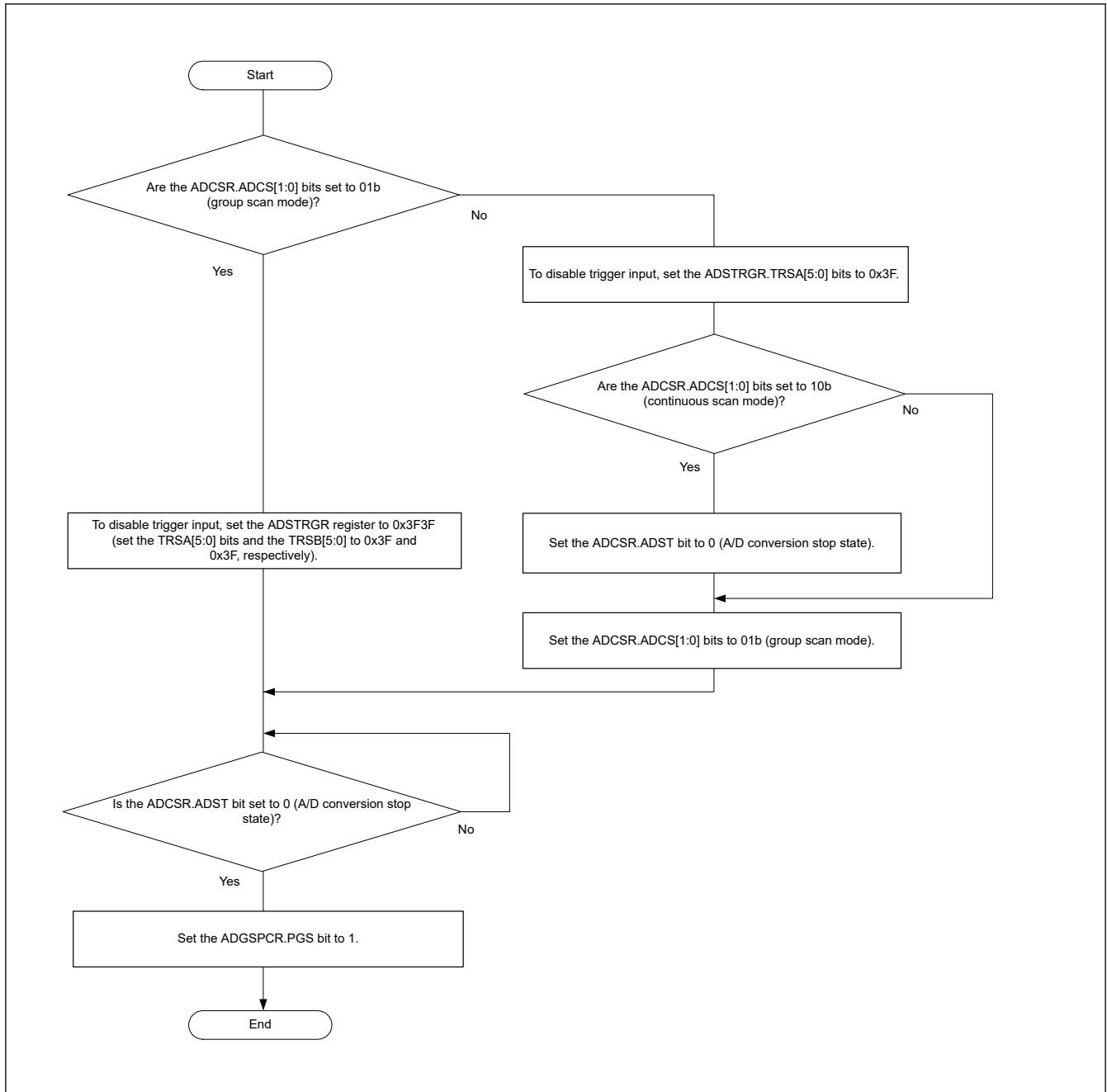


Figure 42.26 Flowchart for ADGSPCR.PGS bit setting

**Table 42.26 Control of A/D conversion operations according to ADGSPCR.GBRSCN bit setting**

A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion for group B is performed after A/D conversion for group A completes.
When A/D conversion for group B is in progress	Input of trigger for group A	A/D conversion for group B is discontinued and A/D conversion for group A starts.	<ul style="list-style-type: none"> <li>A/D conversion for group B is discontinued and A/D conversion for group A starts.</li> <li>A/D conversion for group B starts after A/D conversion for group A completes.</li> </ul>
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.

To use group priority operation mode, select the operation mode to be implemented and set the registers according to the following table.

**Table 42.27 Group priority operation setting and operation mode for two groups (ADGSPCR.PGS = 1)**

ADGSPCR			Operation category
GBRSCN	LGRRS	GBRP	
0	x	0	Group priority operation for two groups (groups A and B) <ul style="list-style-type: none"> <li>When a trigger of group A is input, A/D conversion for group B is terminated (and will not be restarted).</li> </ul>
1	0	0	Group priority operation for two groups (groups A and B) <ul style="list-style-type: none"> <li>After A/D conversion for group B stopped, when A/D conversion for group A completes, A/D conversion for the group B channels selected in the ADANSB0 register restarts according to the conversion order of smaller channel number.</li> </ul>
1	1	0	Group priority operation for two groups (groups A and B) <ul style="list-style-type: none"> <li>After A/D conversion for group B stopped, when A/D conversion for group A completes, A/D conversion for the group B channels selected in the ADANSB0 register restarts according to the conversion order of smaller channel number, beginning from the channel for which A/D conversion stopped.*1</li> </ul>
x	0	1	Group priority operation for two groups (groups A and B) <ul style="list-style-type: none"> <li>Single scanning for group B is continuously performed without a start trigger input. After A/D conversion for group B stopped, when A/D conversion for group A completes, single scanning for the channels selected in the ADANSB0 register restarts according to the conversion order of smaller channel number.</li> </ul>
1	1	1	Group priority operation for two groups (groups A and B) <ul style="list-style-type: none"> <li>Single scanning for group B is continuously performed without a start trigger input. After A/D conversion for group B stopped, when A/D conversion for group A completes, single scanning for the channels selected in the ADANSB0 register restarts according to the conversion order of smaller channel number, beginning from the channel for which A/D conversion stopped.*1</li> </ul>

Note: x: Don't care.

Note 1. When the self-diagnosis function is enabled (ADCER.DIAGM = 1), A/D conversion for the channel that has been stopped is started after self-diagnosis is performed.

### (1) Group priority operation for two groups (when ADGSPCR.PGS = 1)

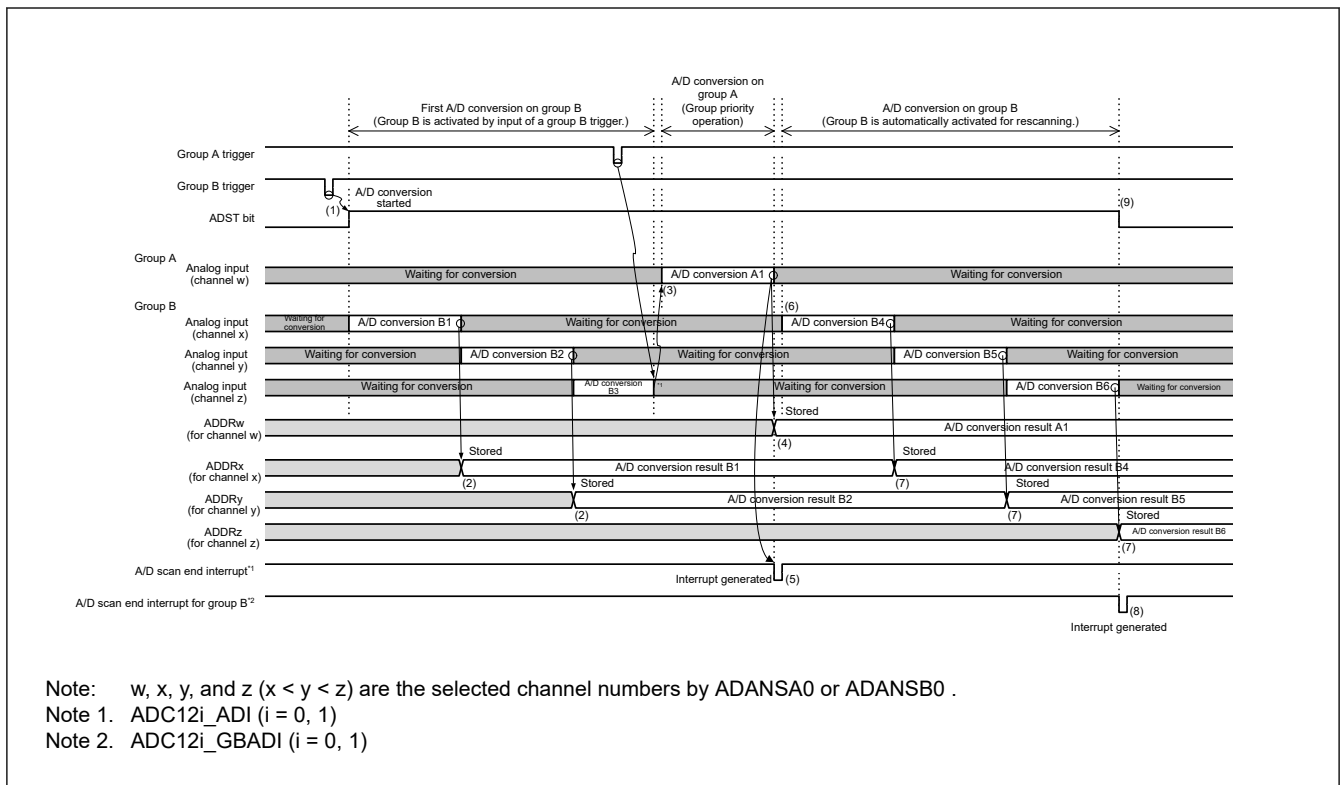
Operation examples 1-1 to 1-3 show group priority operations in group-scan mode (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0).

#### Operation example 1-1: "Group A trigger input during group B scan" when rescanning is enabled

- When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSB0 register starts according to the conversion order from the channel with the smallest number n.
- On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
- When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1. Then A/D conversion for the group A analog input channels selected in the ADANSA0

register starts according to the conversion order from the channel with the smallest number n. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).

4. On completion of A/D conversion on the channels, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. An ADC120\_ADI interrupt request is generated.
6. If the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation), A/D conversion for the group B analog input channels selected in the ADANSB0 register restarts according to the conversion order from the channel with the smallest number n while the ADCSR.ADST remains 1.
7. On completion of A/D conversion on the channels, the result is stored in the corresponding A/D Data Register y (ADDRy).
8. If the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan), a group B scan end interrupt request is generated.
9. When A/D conversion for all the channels completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state.



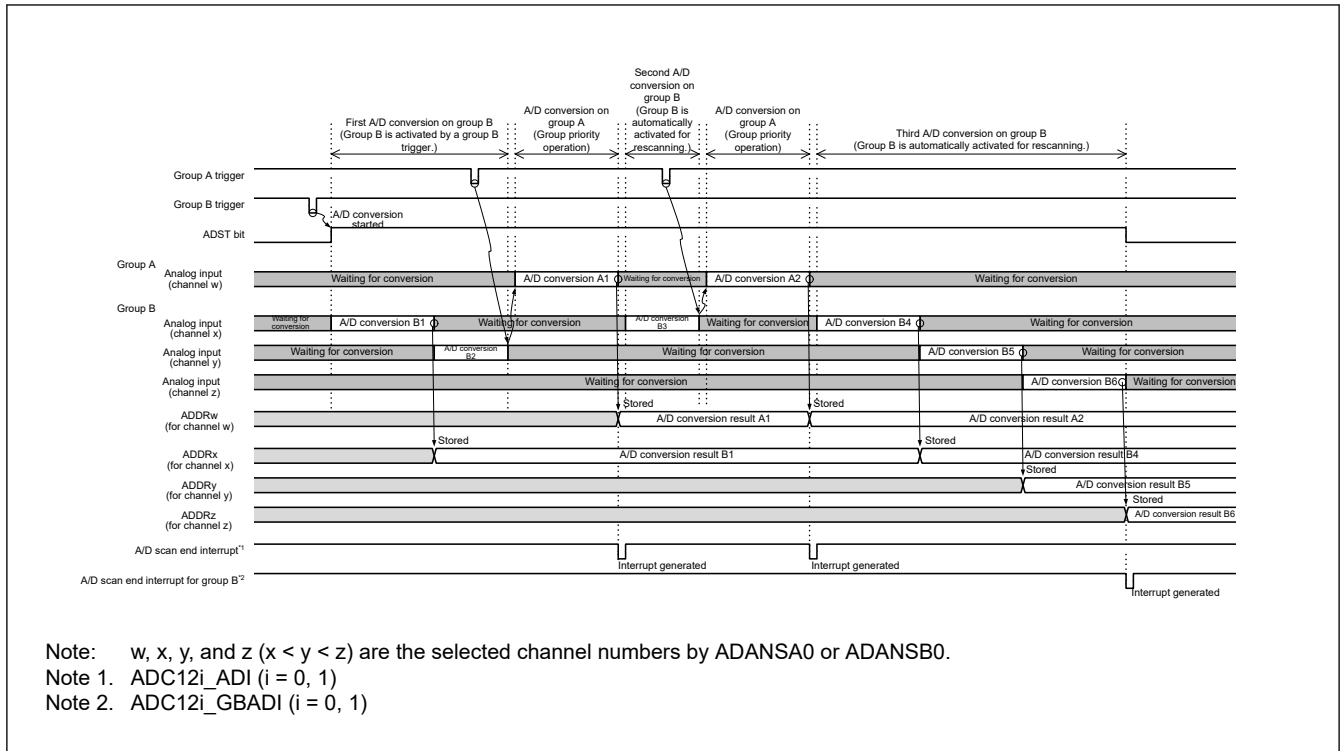
**Figure 42.27 Example of group priority operation 1-1: Group A trigger input during group B scanning when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)**

**Operation example 1-2: “Group A trigger input during rescanning of group B” when rescanning is enabled**

Figure 42.28 shows the operation when a group A trigger is input during rescanning operation for group B.

Even during rescanning operation, when a trigger for group A is input, A/D conversion on group B stops and A/D conversion for group A starts. A/D conversion for group B starts after A/D conversion for group A completes.

Operations for setting the ADCSR.ADST bit, storing the A/D conversion result in the corresponding A/D Data Register y (ADDRy), and generating interrupt requests are the same as those in operation example 1-1.

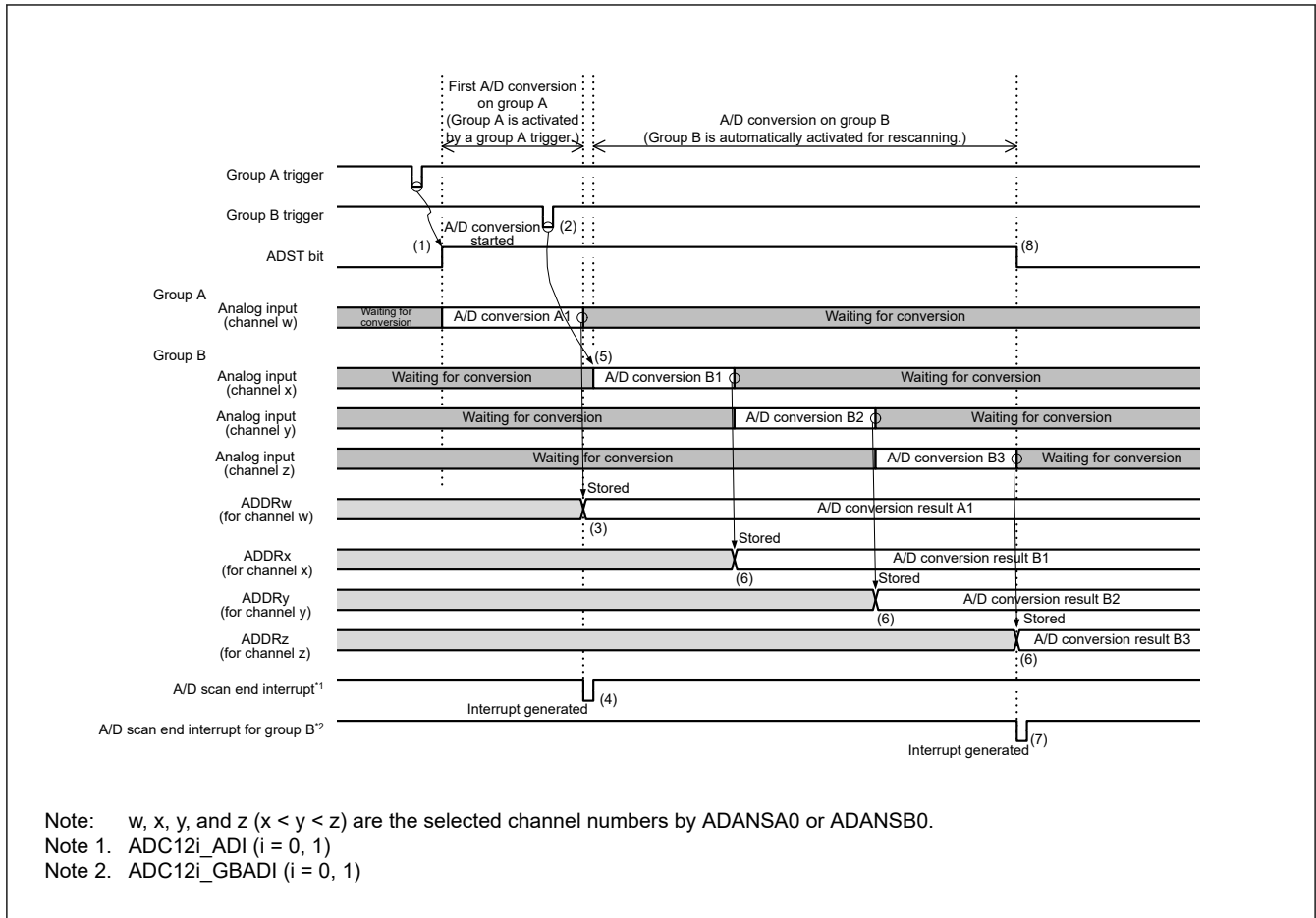


**Figure 42.28 Example of group priority operation 1-2: Group A trigger input during rescanning of group B when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)**

**Operation example 1-3: “Group B trigger input during group A scan” when rescanning is enabled**

The following describes the operation when the setting of the ADGSPCR.GBRSCN bit is 1 (enabling rescanning of the group that was stopped in group priority operation) and a trigger for group B is input during scanning operation for group A. If the setting of the ADGSPCR.GBRSCN bit is 0, any trigger for group B that is input during scanning operation for group A is invalid.

1. When input of a trigger for group A sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the group A analog input channels selected in the ADANSA0 register starts according to the conversion order from the channel with the smallest number n.
2. When a trigger for group B is input during A/D conversion for group A, group B is ready for A/D conversion.
3. On completion of A/D conversion for each channel in group A, the result is stored in the corresponding A/D Data Register y (ADDRy).
4. An ADC120\_ADI interrupt request is generated.
5. When A/D conversion for group A completes, while the ADCSR.ADST bit remains 1, A/D conversion for the group B analog input channels selected in the ADANSB0 register starts according to the conversion order from the channel with the smallest number n.  
 As with the case of operation example 1-1, if a trigger for group A is input during A/D conversion for group B, A/D conversion for group A starts. Then A/D conversion for group B starts upon completion of A/D conversion for group A.
6. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
7. Upon completion of A/D conversion for group B, a group B scan end interrupt request is generated if the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan).
8. When A/D conversion for all the channels completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state.



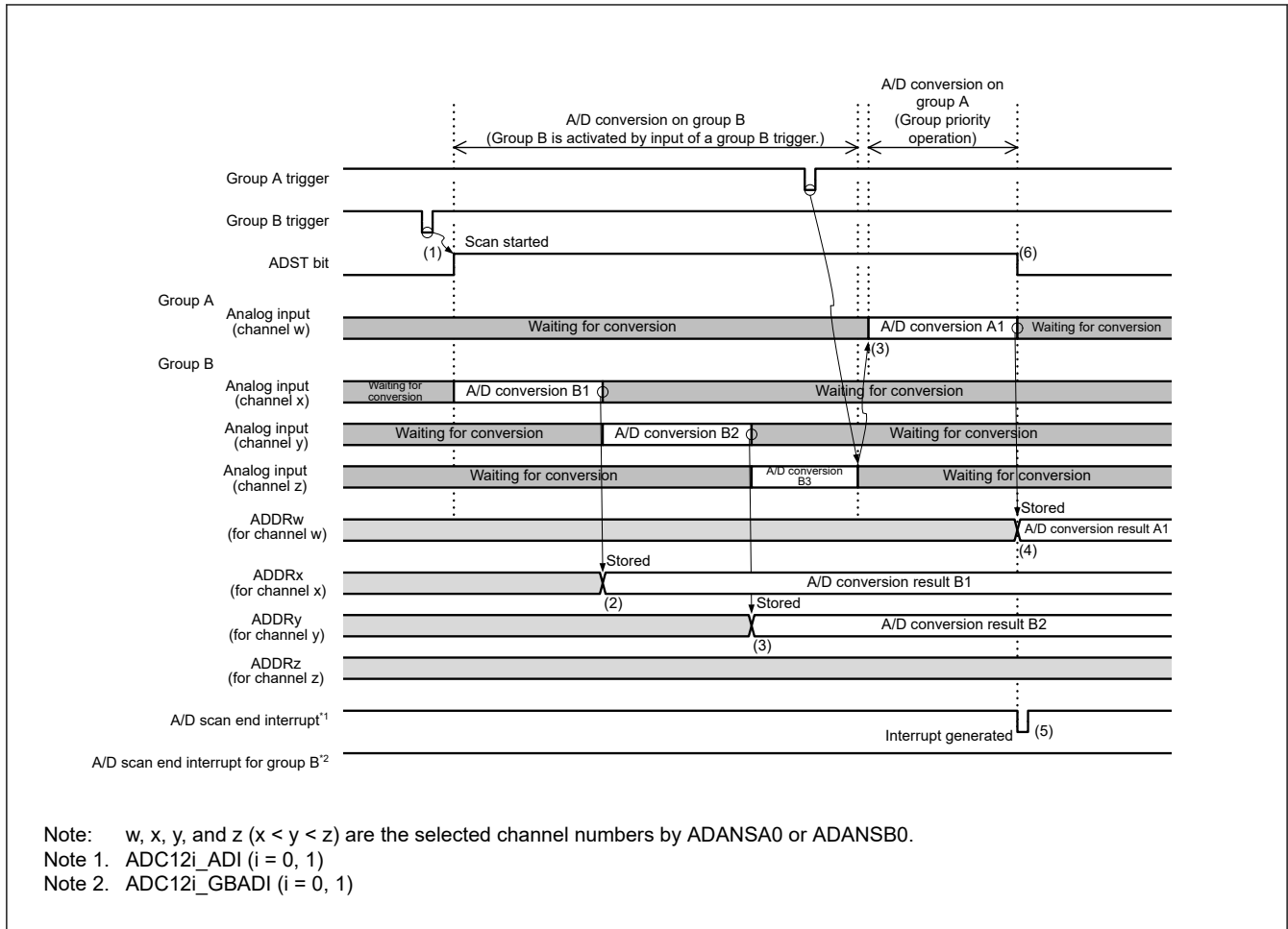
**Figure 42.29 Example of group priority operation 1-3: Group B trigger input during group A scan when rescanning is enabled (when ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)**

Operation example 1-4 shows the group priority operation in group-scan mode (when ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0).

**Operation example 1-4: “Group A trigger input during group B scan” when rescanning is disabled**

1. When input of a trigger for group B sets the ADCSR.ADST bit to 1 (starting A/D conversion), A/D conversion for the analog input channels selected in the ADANSB0 register starts according to the conversion order from the channel with the smallest number n.
2. On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for the group A analog input channels selected in the ADANSA0 register starts according to the conversion order from the channel with the smallest number n. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
4. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. On completion of A/D conversion for group A, an ADC120\_ADI interrupt request is generated.
6. When A/D conversion for group A completes, the ADCSR.ADST bit is automatically cleared and the A/D converter enters a wait state. A/D conversion for group B is not performed until a trigger for group B is input the next time.





**Figure 42.30 Group priority operation example 1-4: “Group A trigger is input during group B scan” when rescanning is disabled (when ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, and ADGSPCR.LGRRS = 0)**

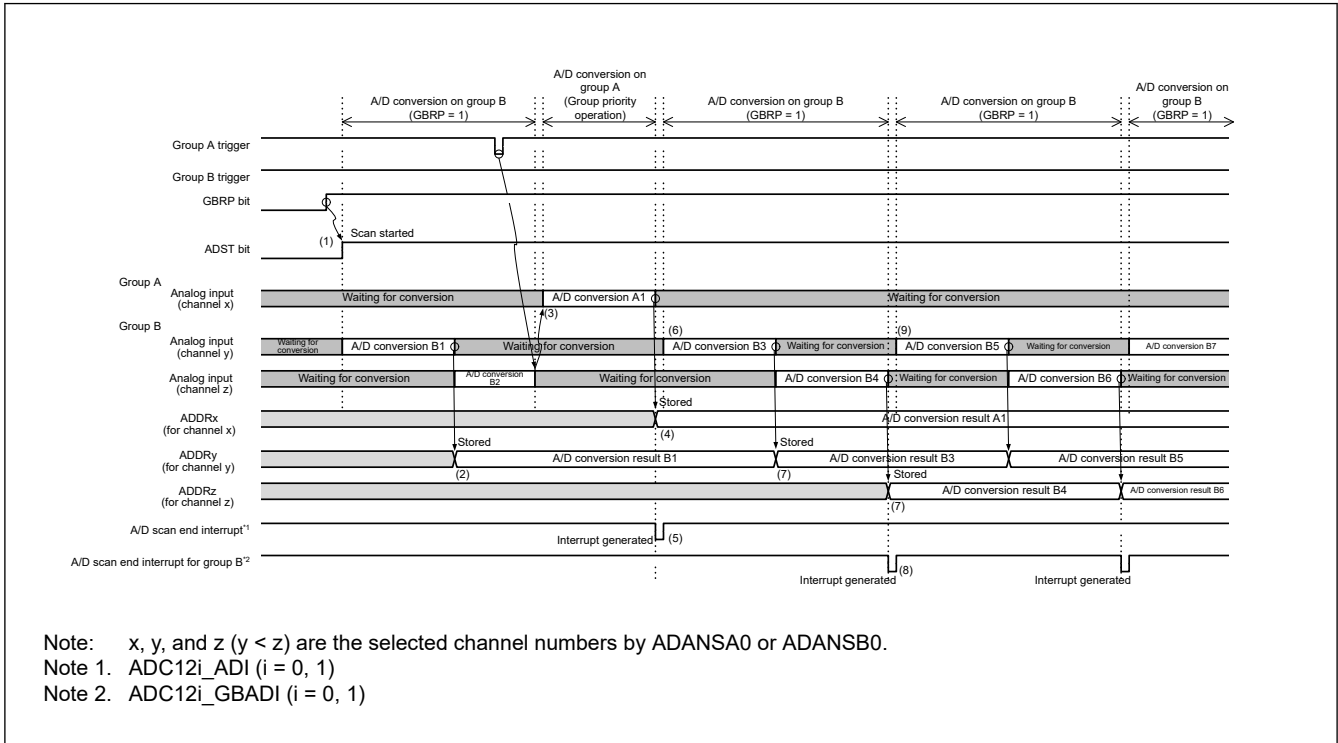
Operation example 1-5 shows the group priority operation in group-scan mode (when ADGSPCR.GBRP = 1, and ADGSPCR.LGRRS = 0).

**Operation example 1-5: Continuously activating single-scan operation for group B**

1. When ADGSPCR.GBRP = 1 is set, the ADCSR.ADST bit is set to 1 (starting A/D conversion) and A/D conversion for the analog input channels selected in the ADANSB0 register starts according to the conversion order from the channel with the smallest number n.
2. On completion of A/D conversion for each channel in group B, the result is stored in the corresponding A/D Data Register y (ADDRy).
3. When a trigger for group A is input during A/D conversion for group B, A/D conversion for group B stops while the ADCSR.ADST bit remains 1, and then A/D conversion for group A analog input channels selected in the ADANSA0 register starts according to the conversion order from the channel with the smallest number n. If A/D conversion stops before it is completed, the conversion result is not stored in the A/D Data Register y (ADDRy).
4. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).
5. On completion of A/D conversion for group A, an ADC120\_ADI interrupt request is generated.
6. If ADGSPCR.GBRP = 1 is set (performing single scan continuously), A/D conversion for the group B analog input channels selected in the ADANSB0 register restarts according to the conversion order from the channel with the smallest number n while the ADCSR.ADST remains 1 (starting A/D conversion).
7. On completion of A/D conversion of a single channel, the result is stored in the corresponding A/D Data Register y (ADDRy).

8. If the setting of the ADCSR.GBADIE bit is 1 (enabling interrupt generation on completion of group B scan), a group B scan end interrupt request is generated.
9. If ADGSPCR.GBRP = 1 is set (performing single scan continuously), A/D conversion for the group B analog input channels selected in the ADANSB0 register restarts according to the conversion order from the channel with the smallest number n while the ADCSR.ADST remains 1 (starting A/D conversion).

Steps 6 to 9 are repeated as long as the ADGSPCR.GBRP bit remains 1. Do not clear the ADCSR.ADST bit as long as the ADGSPCR.GBRP bit is 1. To forcibly stop A/D conversion while ADGSPCR.GBRP = 1, follow the procedure shown in Figure 42.43.



**Figure 42.31 Group priority operation example 1-5: Continuously activating single scan for group B (when ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 0)**

Note: To continuously activate single-scan operation for group B, disable group B trigger input.

### 42.3.5 Compare Function for Windows A and B

#### 42.3.5.1 Compare Function Windows A and B

The compare function compares a reference value with the A/D conversion result. The reference value can be set for Window A and Window B independently. When the compare function is in use, the self-diagnosis function and double trigger mode cannot be used. The main differences between Window A and Window B are their different interrupt output signals and the constraint on Window B of only one selectable channel.

This section provides an example operation that combines continuous scan mode and the compare function.

The operation is as follows:

1. When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, a synchronous trigger (GPT, ELC) or an asynchronous trigger, A/D conversion starts in the order of the selected channels, temperature sensor, and internal reference voltage.
2. On completion of A/D conversion, the A/D conversion result is stored in the associated A/D Data Register y (ADDRy, ADTSDR, or ADOCDR). When ADCMPCR.CMPAE = 1, if bits in the ADCMPANSR0 register or the ADCMPANSER register are set for Window A, the A/D conversion result is compared with the set ADCMPDR0/1 register value. When ADCMPCR.CMPBE = 1, if bits in the ADCMPBNSR register are set for Window B, the A/D conversion result is compared with the ADWINULB/ADWINLLB register setting.

3. As a result of the comparison, when Window A meets the condition set in ADCMPLR0/1 or ADCMPLER, the Compare Window A Flag (ADCMPSTR0.CMPSTCHAN, ADCMPSTR.CMPSTTSA or ADCMPSTR.CMPSTOCA) sets 1. At this time, if the ADCMPSTR.CMPAIE bit is 1, an ADC12i\_CMPAI (i = 0, 1) interrupt request is generated. In the same way, when Window B meets the condition set in ADCMPBSR.CMPLB, the Compare Window B Flag (ADCMPBSR.CMPSTB) sets to 1. At this time, if the ADCMPSTR.CMPBIE bit is 1, an ADC12i\_CMPBI (i = 0, 1) interrupt request is generated.
4. On completion of all selected A/D conversions and comparisons, scan restarts.
5. After the ADC12i\_CMPAI (i = 0, 1) and ADC12i\_CMPBI (i = 0, 1) interrupts are accepted, the ADCSR.ADST bit is set to 0 (A/D conversion stop) and processing is performed for channels for which the compare flag is set to 1.
6. When all compare flags of Window A are cleared, the ADC12i\_CMPAI (i = 0, 1) interrupt request is canceled. In the same way, when all compare flags of Window B are cleared, the ADC12i\_CMPBI (i = 0, 1) interrupt request is reset. To perform comparison again, restart the A/D conversion.

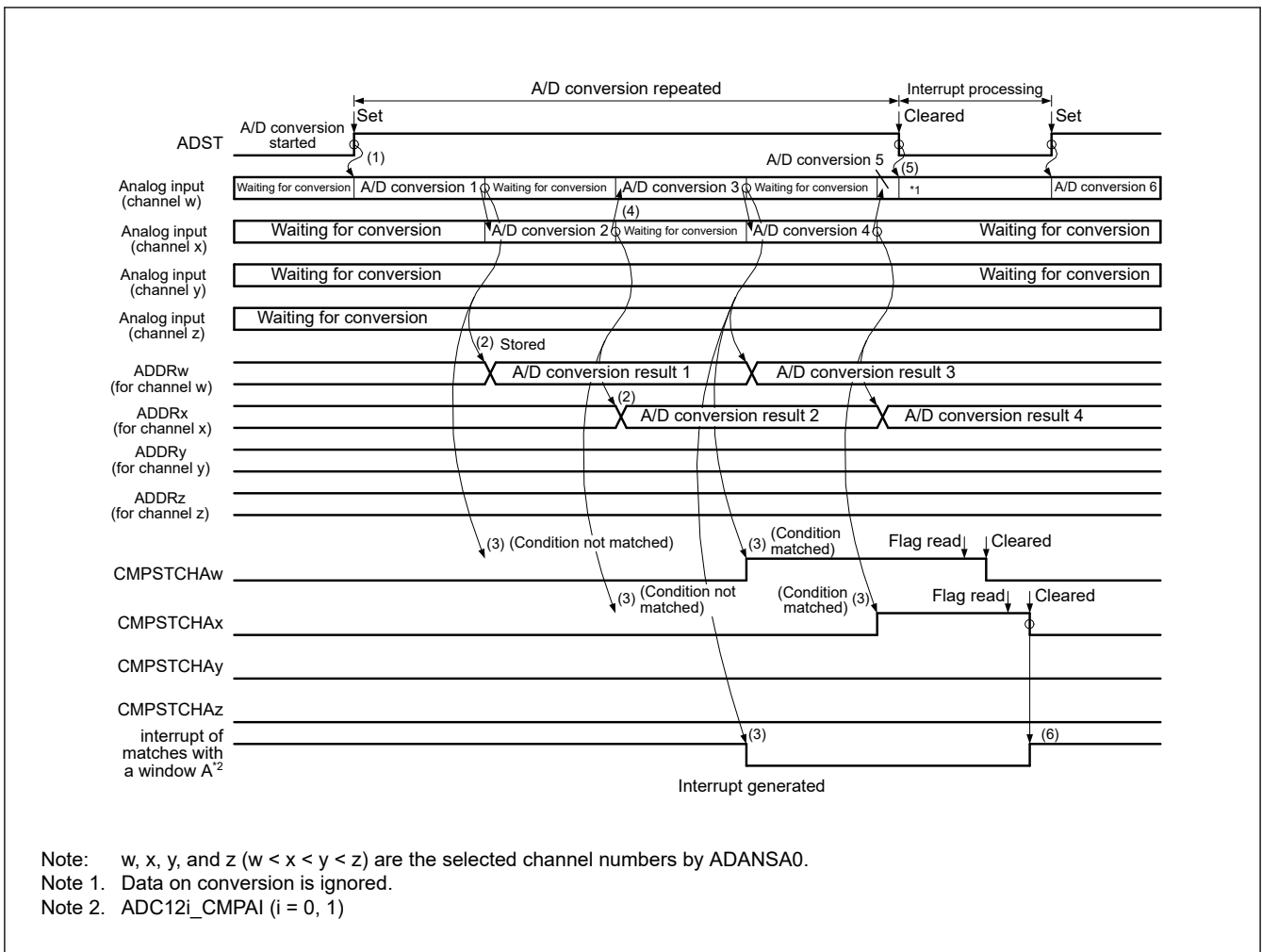


Figure 42.32 Example of compare function operation, when the analog inputs (channel w to z) are compared

### 42.3.5.2 Event Output of Compare Function

The event output of the compare function specifies the upper-side reference voltage value and the lower-side reference voltage value for window A and window B, respectively. The output compares the A/D converted value of the selected channel with the upper and lower side reference voltage value and outputs events (ADC12i\_WCMPLM (i = 0, 1)/ADC12i\_WCMPUM (i = 0, 1)) based on event conditions (A OR B, A AND B, A XOR B) and comparison result of window A and window B.

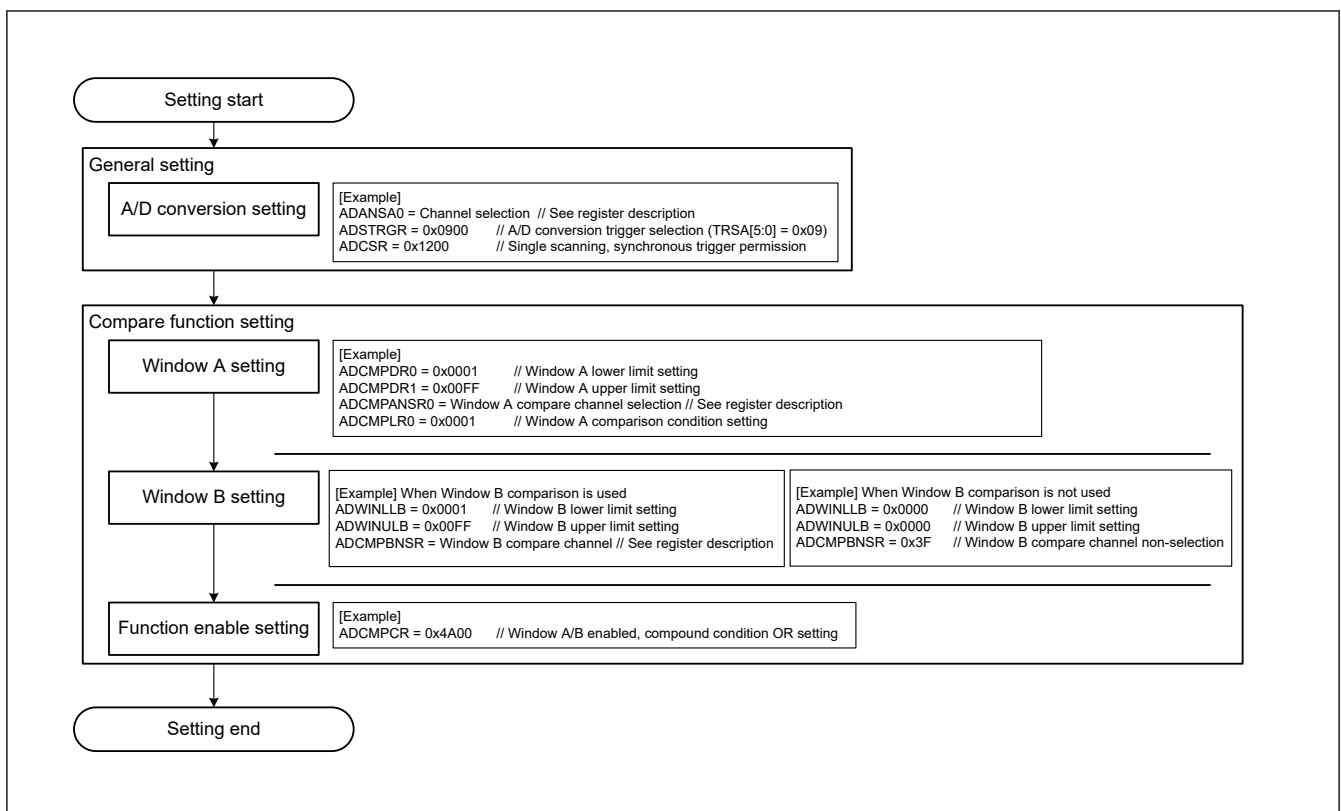
If more than one channel is selected for window A, and even when one channel in window A meets the comparison condition, the comparison result of window A is met. When using this function, perform A/D conversion in single scan mode.

Any channels from analog input, internal reference voltage, temperature sensor output and VBATT 1/3 voltage monitor output are selectable for window A.

One channel from analog input, internal reference voltage, temperature sensor output and VBATT 1/3 voltage monitor output is selectable for window B.

The following sequence is an example of how to set up and use the event output of the compare function:

1. Confirm that the value in the ADCSR.ADCS bits is 00b (single scan mode).
2. Select the channel for window A in the ADCMPANSR0 and ADCMPANSER registers. Set the window comparison conditions in the ADCMPLR0/1 and ADCMPLE registers. Set the upper-side and lower-side reference values in the ADCMPDR0/1 registers.
3. Select the channel and comparison conditions for Window B in the ADCMPBNSR register, and set the upper and lower reference values in the ADWINULB and ADWINLLB registers.
4. Set the composite conditions for window A/B, window A/B operation enable, and interrupt output enable in the ADCMPCR register.



**Figure 42.33 Setting example when using the event output of the compare function**

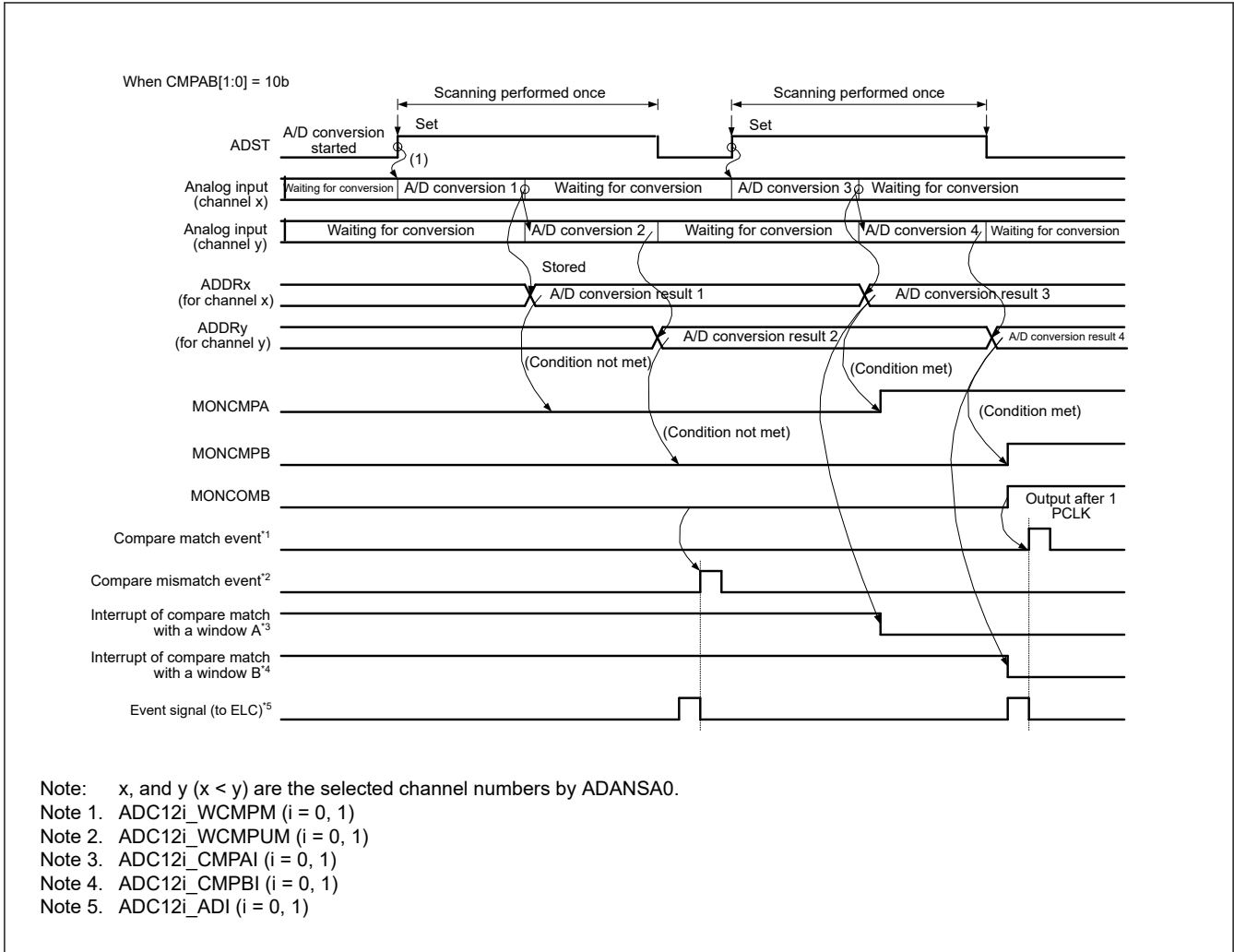
For event output usage when using only window A for the compare function, note the following:

- Enable both Window A and Window B (ADCMPCR.CMPAE = 1, ADCMPCR.CMPBE = 1)
- Set the compound condition of Window A and Window B to “OR condition” (ADCMPCR.CMPAB[1:0] = 00b)
- Set the compared channel of Window B to “No selection” (ADCMPBNSR.CMPCHB[5:0] = 0x3F)
- Set the compare condition of Window B to “0 < results < 0 always means mismatch”. (ADCMPCR.WCMPE = 1, ADWINLLB[15:0] = ADWINULB[15:0] = 0x0000, and ADCMPBNSR.CMPLB = 1)

Figure 42.34 shows the event output operation example of compare function.

A scan end event (ADC12i\_ADI (i = 0, 1)) is output with the same timing as single scan completion. A match or mismatch event (ADC12i\_WCMPM (i = 0, 1)/ADC12i\_WCMPUM (i = 0, 1)) is output with 1 PCLKA cycle delay depending on the ADCMPCR.CMPAB[1:0] settings.

Note: The match and mismatch events are exclusive, so both events are never output simultaneously.



**Figure 42.34 Example operation of the compare function event output, when the analog inputs (channel x and y) are compared**

Note: Event output of compare function outputs match/mismatch from the comparison results of Window A and Window B, based on the ADCMPCR.CMPAB[1:0] settings.

Note: The comparison result of Window A is the logical addition of the comparison results of the comparison target channels of Window A. The comparison results of Window A and Window B are updated by each A/D conversion, and are kept even when single scan ends. Set ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0 to clear the comparison results to 0.

### 42.3.5.3 Restrictions on Compare Function

The following constraints apply for the compare function:

- The compare function cannot be used together with the self-diagnosis function or double-trigger mode. The compare function is not available for ADDR, ADDBLDR, ADDBLDRA, and ADDBLDRB.
- Specify single scan mode when using match/mismatch event outputs.
- When the temperature sensor output, internal reference voltage is selected for Window A, Window B operations are prohibited.
- When the temperature sensor output, internal reference voltage is selected for Window B, Window A operations are prohibited.
- Setting the same channel for Window A and Window B is prohibited.
- Specify single scan mode when using the buffer function. It is prohibited to use double trigger mode together.
- Set the reference voltage values so that the high-potential reference voltage value is equal to or larger than the low potential reference voltage value.

### 42.3.6 Analog Input Sampling and Scan Conversion Time

Scan conversion can be activated either by a software trigger, a synchronous trigger (GPT, ELC), or an asynchronous trigger (ADTRGn (n = 0, 1)). After the start-of-scanning-delay time ( $t_D$ ) has elapsed, processing by the channel-dedicated sample-and-hold circuits, processing for disconnection detection assistance, and processing of conversion for self-diagnosis all proceed, followed by processing for A/D conversion.

Figure 42.35 shows the scan conversion timing, in which scan conversion is activated by a software trigger or a synchronous trigger (GPT, ELC). Figure 42.36 shows the scan conversion timing, in which scan conversion is activated by an asynchronous trigger (ADTRGn (n = 0, 1)). The scan conversion time ( $t_{SCAN}$ ) includes the start-of-scanning-delay time ( $t_D$ ), channel-dedicated sample-and-hold circuit processing time ( $t_{SPLSH}$ )<sup>\*1</sup>, disconnection detection assistance processing time ( $t_{DIS}$ )<sup>\*2</sup>, self-diagnosis A/D conversion processing time ( $t_{DIAG}$  and  $t_{DSD}$ )<sup>\*3</sup>, A/D conversion processing time ( $t_{CONV}$  and  $t_{DSD}$ ), channel-dedicated sample-and-hold circuit end time ( $t_{SHED}$ )<sup>\*4</sup> and end-of-scanning-delay time ( $t_{ED}$ ).

The A/D conversion processing time ( $t_{CONV}$ ) consists of input sampling time ( $t_{SPL}$ ) and time for conversion by successive approximation ( $t_{SAM}$ ). The sampling time ( $t_{SPL}$ ) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTRn register.

The time for conversion by successive approximation ( $t_{SAM}$ ) is the following:

- 13 ADCLK states with 12-bit accuracy selected.
- 11 ADCLK states with 10-bit accuracy selected.
- 9 ADCLK states with 8-bit accuracy selected.

Table 42.28 shows the time for conversion by successive approximation ( $t_{SAM}$ ).

The scan conversion time ( $t_{SCAN}$ ) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + t_{ED} + (t_{CONV} \times n) + t_{ED}^{*5}$$

The scan conversion time for the first cycle in continuous scan mode is  $t_{SCAN}$  for single scan minus  $t_{ED}$ . The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed in the following:

$$t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + t_{DSD} + t_{SHED} + (t_{CONV} \times n) + t_{SHED}^{*5}$$

Note 1. When no channel-dedicated sample-and-hold circuits are used,  $t_{SPLSH} = 0$ .

Note 2. When disconnection detection assistance is not selected,  $t_{DIS} = 0$ .

Note 3. When the self-diagnosis function is not used,  $t_{DIAG} = 0$ ,  $t_{DSD} = 0$ .

Note 4. When no channel-dedicated sample-and-hold circuits are used,  $t_{SHED} = 0$ , assuming continuous scan mode is active. In single scan mode and group scan mode,  $t_{SHED}$  is included in the end-of-scanning-delay ( $t_{ED}$ ).

Note 5. When input sampling times ( $t_{SPL}$ ) of all selected channels are the same, this element equals  $t_{CONV} \times n$ . If each channel has a different sampling time, this element equals that of  $t_{SPL}$  and  $t_{SAM}$  set to each selected channel.

Table 42.28 shows the times for conversion during scanning.

**Table 42.28 Conversion times during scanning (in numbers of cycles of ADCLK and PCLKA)**

Item			Symbol	Type/Conditions			Unit
				Synchronous trigger <sup>*4</sup>	Asynchronous trigger	Software trigger	
Scan start processing time <sup>*1 *2</sup>	A/D conversion on group A under group A priority control.	Group B is to be stopped (Group A is activated after group B is stopped by of an A/D conversion source from group A).	$t_D$	3 PCLKA + 6 ADCLK 5 PCLKA + 3 ADCLK <sup>*5</sup>	—	—	Cycles
		Group B is not to be stopped (activation by an A/D conversion source from group A).		2 PCLKA + 4 ADCLK	—	—	
	A/D conversion when self-diagnosis is enabled.	A/D conversion for self-diagnosis is to be started.		2 PCLKA + 6 ADCLK	4 PCLKA + 6 ADCLK	6 ADCLK	
	All other			2 PCLKA + 4 ADCLK	2 PCLKA + 4 ADCLK	4 ADCLK	
Channel-dedicated sample-and-hold processing time <sup>*1</sup>	Sampling time		$t_{SPLSH}$	$t_{SH}$	Without continuous sampling: setting in ADSHCR.SSTSH[7:0] (initial value = 18h × ADCLK) With continuous sampling: 0		
	Wait time between sampling and A/D Conversion				$t_w$	12	
Disconnection detection assistance processing time			$t_{DIS}$	Setting in ADNDIS[3:0] (initial value = 0x0) × ADCLK			
Self-diagnosis conversion processing time <sup>*1</sup>	Sampling time		$t_{DIAG}$	$t_{SPL}$	Setting in ADSSTR00 (initial value = 0x0B) × ADCLK <sup>*3</sup>		
	Time for conversion by successive approximation	12-bit conversion accuracy			$t_{SAM}$	15 ADCLK	
		10-bit conversion accuracy		13 ADCLK			
		8-bit conversion accuracy		11 ADCLK			
	Wait time between self-diagnosis conversion end and analog channel sampling start.			$t_{DED}$	2 ADCLK		
Wait time between last channel conversion end and self-diagnosis sampling start in continuous scan mode.		$t_{DSD}$	2 ADCLK				
A/D conversion processing time <sup>*1</sup>	Sampling time		$t_{CONV}$	$t_{SPL}$	Setting in ADSSTRn (n = 0 to 8 (unit 0), 0 to 5 (unit 1), T, O, V) (initial value = 0x0B) × ADCLK + 0.5 ADCLK		
	Time for conversion by successive approximation	12-bit conversion accuracy			$t_{SAM}$	13 ADCLK	
		10-bit conversion accuracy		11 ADCLK			
		8-bit conversion accuracy		9 ADCLK			
Channel-dedicated sample-and-hold end processing time			$t_{SHED}$	2 ADCLK			
Scan end processing time <sup>*1</sup>			$t_{ED}$	1 PCLKA + 3 ADCLK 2 PCLKA + 3 ADCLK <sup>*5</sup>			

Note 1. See [Figure 42.35](#) and [Figure 42.36](#) for an illustration of times  $t_D$ ,  $t_{SPLSH}$ ,  $t_{DIAG}$ ,  $t_{CONV}$ , and  $t_{ED}$ .

Note 2. This is the maximum time required from software writing or trigger input to A/D conversion start.

Note 3. The sampling time setting must satisfy the electrical characteristics.

Note 4. This does not include the time consumed in the path from timer output to trigger input.

Note 5. If ADCLK is faster than PCLKA (PCLKA to ADCLK frequency ratio = 1:2 or 1:4), the scan end processing time changes.

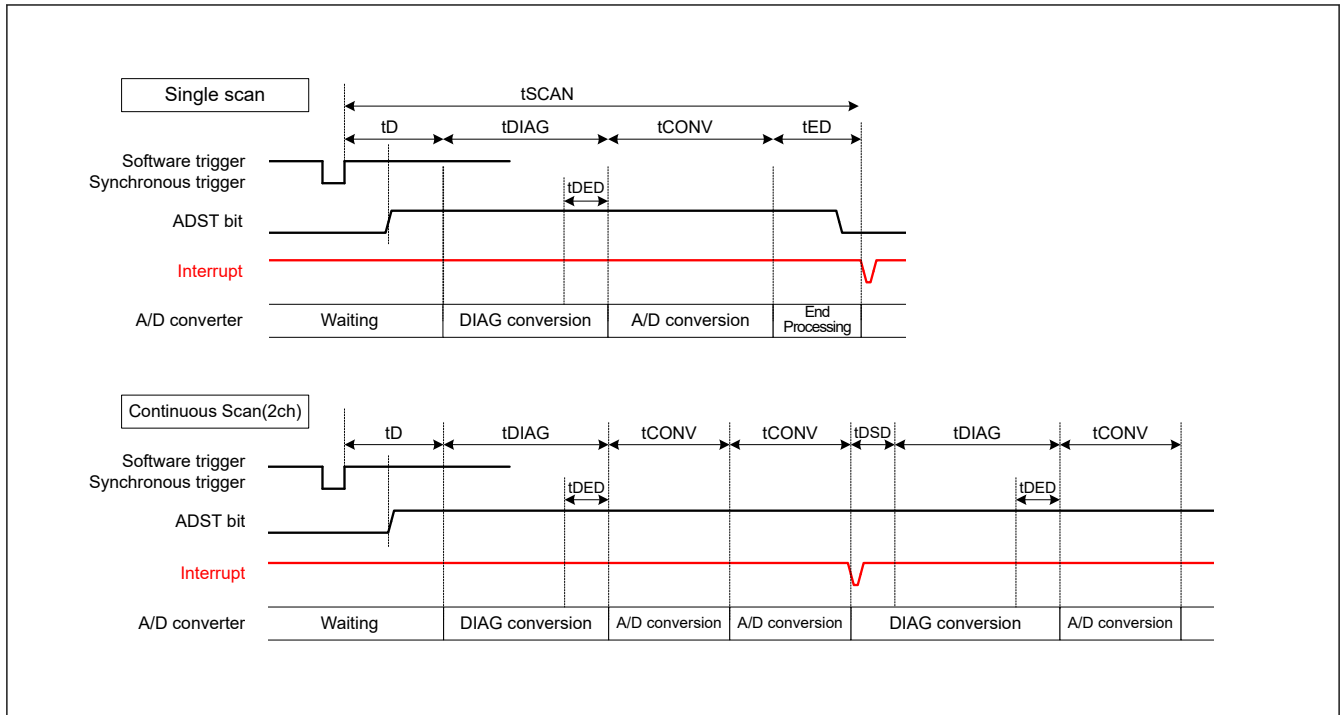


Figure 42.35 Scan conversion timing when activated by software or a synchronous trigger input (GPT, ELC)

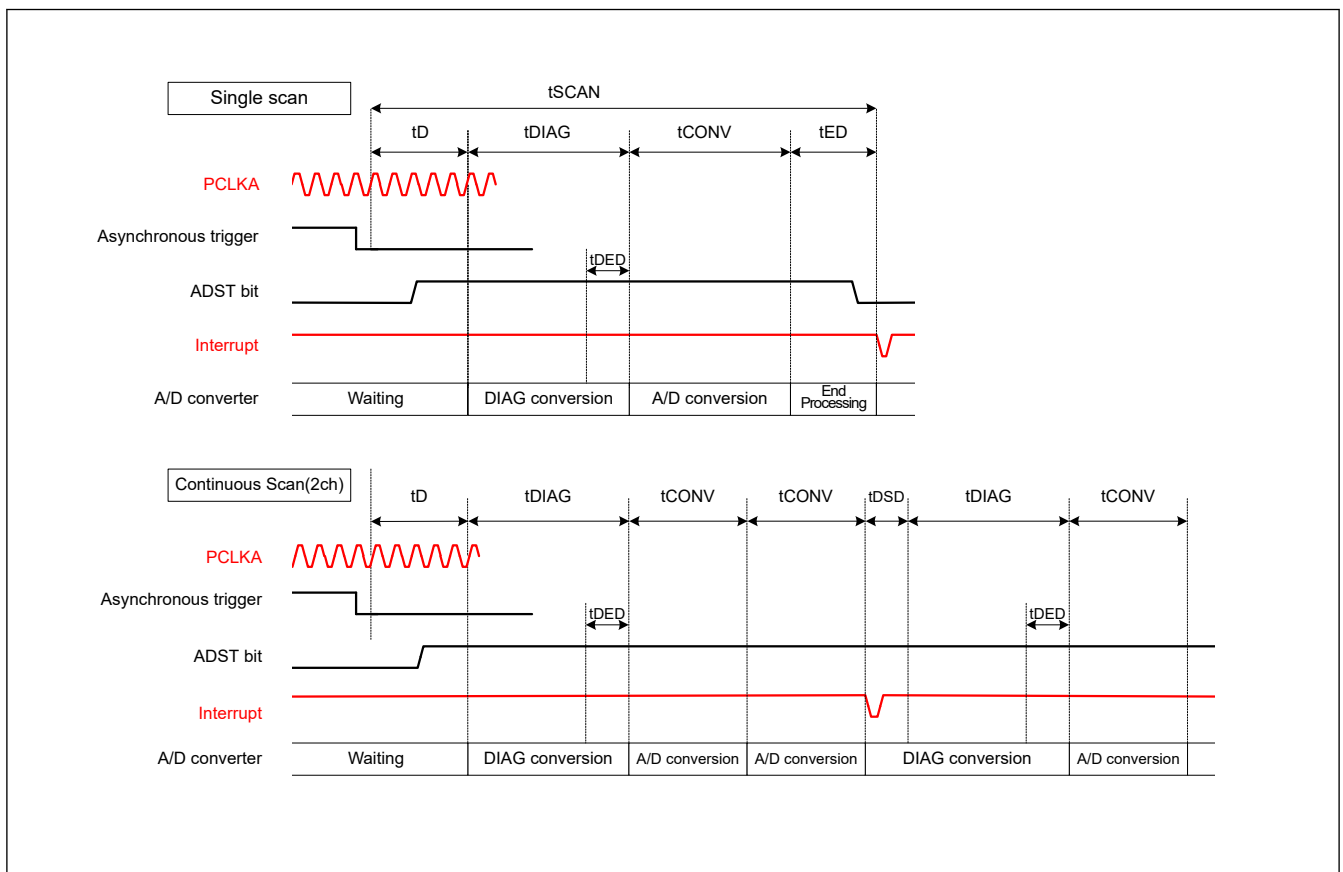


Figure 42.36 Scan conversion timing when activated by an asynchronous trigger input (ADTRGn (n = 0, 1))

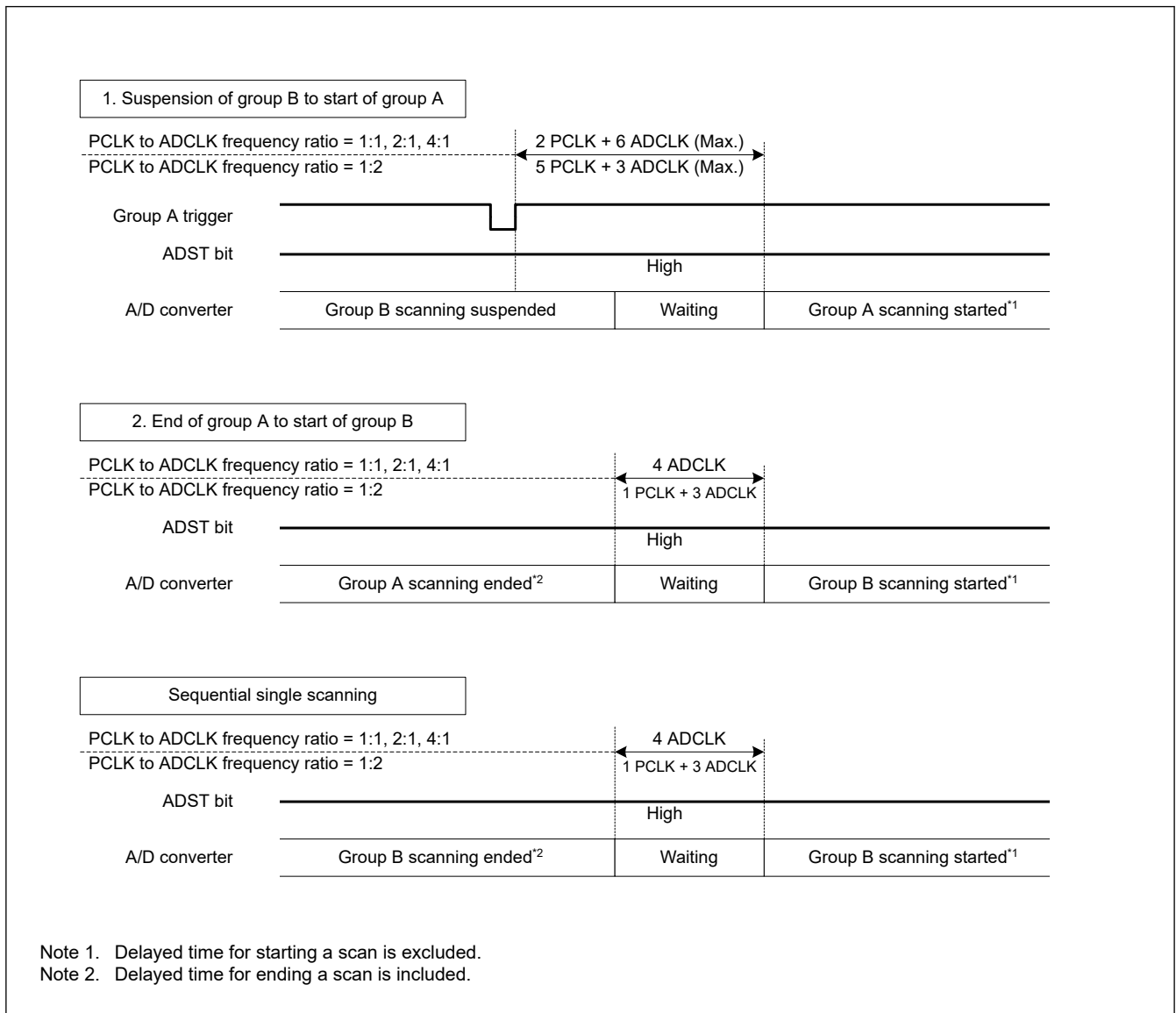


### 42.3.6.1 Timing of Suspension and Starting of Scanning in Operation under Group Priority Control

The timings for suspension and starting of scanning in operation under group priority control that must be considered are listed below.

1. The timing for suspending a scan of a group with a lower-priority and the timing for starting a scan of a group with a higher-priority.
2. The time at which scanning by the group with a lower-priority is resumed on completion of scanning by the higher-priority group when the trigger for scanning by the lower-priority group is accepted during scanning by the higher-priority group.
3. The timing for performing sequential single scans by a lower-priority group.

Figure 1.37 shows the timing diagram of each of the above cases.



**Figure 42.37** Timing diagram of suspension and starting of scanning in operation under group priority control

### 42.3.7 Usage Example of A/D Data Register Automatic Clearing Function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADVMDR) to 0x0000 when the A/D data registers are read by the CPU or DTC or DMAC.

This function enables detection of update failures of the A/D data registers (ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, ADTSDR, ADOCDR, ADVMDR). This section describes examples in which the function to automatically clear the ADDRy register is enabled and disabled.

- If the ADCER.ACE bit is 0 (automatic clearing is disabled) and for some reason, if the A/D conversion result (0x0222) is not written to the ADDRy register, the ADDRy value retains the old data (0x0111). In addition, if this ADDRy value is read into a general-purpose register using an A/D scan end interrupt, the old data (0x0111) can be saved in the general-purpose register. When checking whether there is an update failure, it is necessary to frequently save the old data in SRAM or in a general-purpose register.
- If the ADCER.ACE bit is 1 (automatic clearing is enabled), when ADDRy = 0x0111 is read by the CPU or DTC or DMAC, ADDRy is automatically set to 0x0000. Next, if the A/D conversion result of 0x0222 cannot be transferred to ADDRy for some reason, the cleared data (0x0000) remains as the ADDRy value. If this ADDRy value is read into a general-purpose register using an A/D scan end interrupt, 0x0000 is saved in the general-purpose register. Occurrence of an ADDRy update failure can be determined by checking that the read data value is 0x0000.

### 42.3.8 A/D-Converted Value Addition/Average Mode

A/D-converted value addition/average mode can be used when A/D conversion of the analog input of the selected channels, the temperature sensor output, the internal reference voltage and VBATT 1/3 voltage monitor output is selected.

In A/D-converted value addition mode, the same channel is A/D-converted 1, 2, 3, 4, or 16 consecutive times, and the sum of the converted values is stored in the data register. The conversion count of the addition function can be set to 16 only when 12-bit accuracy is selected. In A/D-converted value average mode, the same channel is A/D-converted 2 or 4 consecutive times, and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.

The A/D-converted value addition/average function can be used when A/D conversion of the analog inputs of the selected channels or A/D conversion of the temperature sensor output or A/D conversion of the internal reference voltage or A/D conversion of the VBATT 1/3 voltage monitor output is selected. The A/D-converted value addition/average function can also be used for channels for which the double-trigger function is selected.

The addition function for self-diagnosis is not provided.

### 42.3.9 Disconnection Detection Assist Function

The ADC12 incorporates a function to fix the charge for sampling capacitance to the specified state (VREFH0 or VREFL0 for unit 0, VREFH or VREFL for unit 1) before the start of A/D conversion. This function enables disconnection detection in wiring of analog inputs.

When using the disconnection detection assist function for the channel-dedicated sample-and-hold circuit, set the ADHMSR.SHMD bit to 0 (select disable continuous sampling function).

Figure 42.38 shows the A/D conversion operation when the disconnection detection assist function is used. Figure 42.39 shows an example of disconnection detection when precharge is selected. Figure 42.40 shows an example of disconnection detection when discharge is selected.

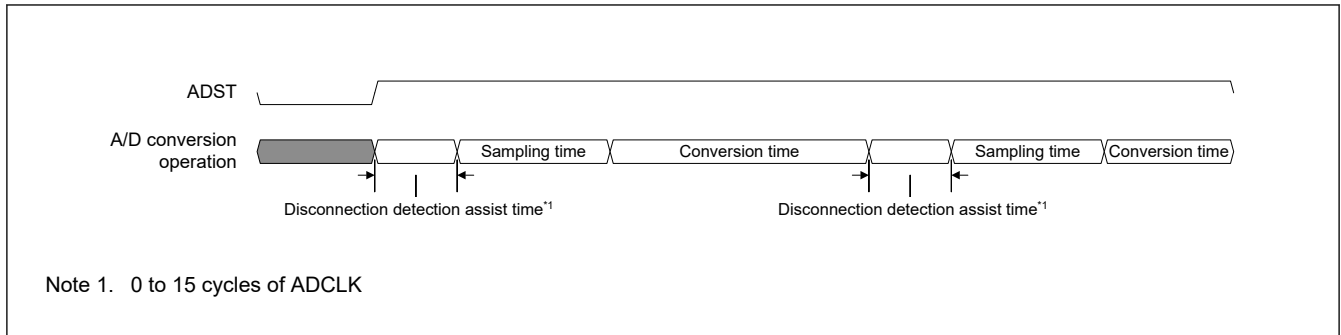
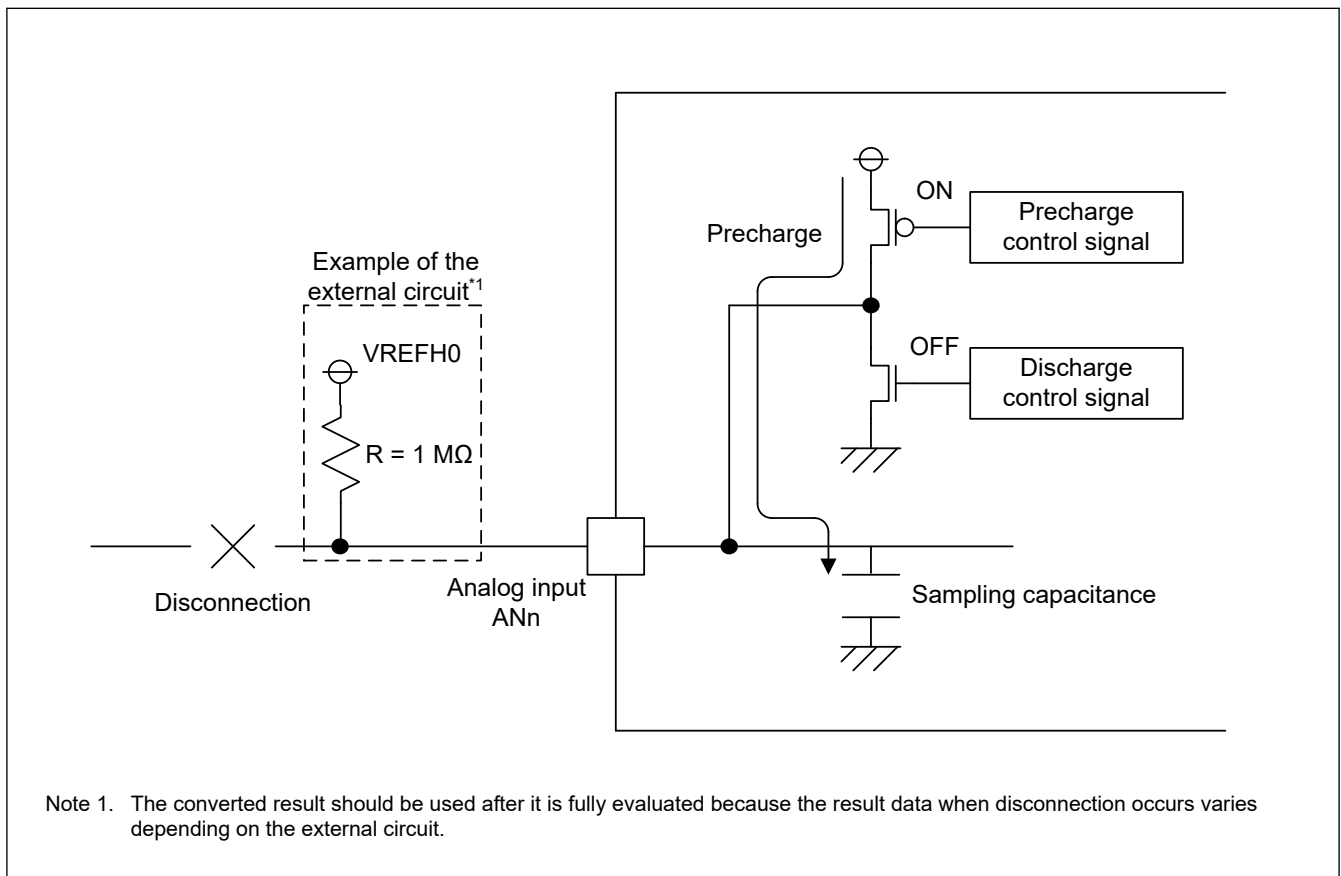


Figure 42.38 Operation of A/D conversion when disconnection detection assist function is used



Note 1. The converted result should be used after it is fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 42.39 Example of disconnection detection when precharge is selected

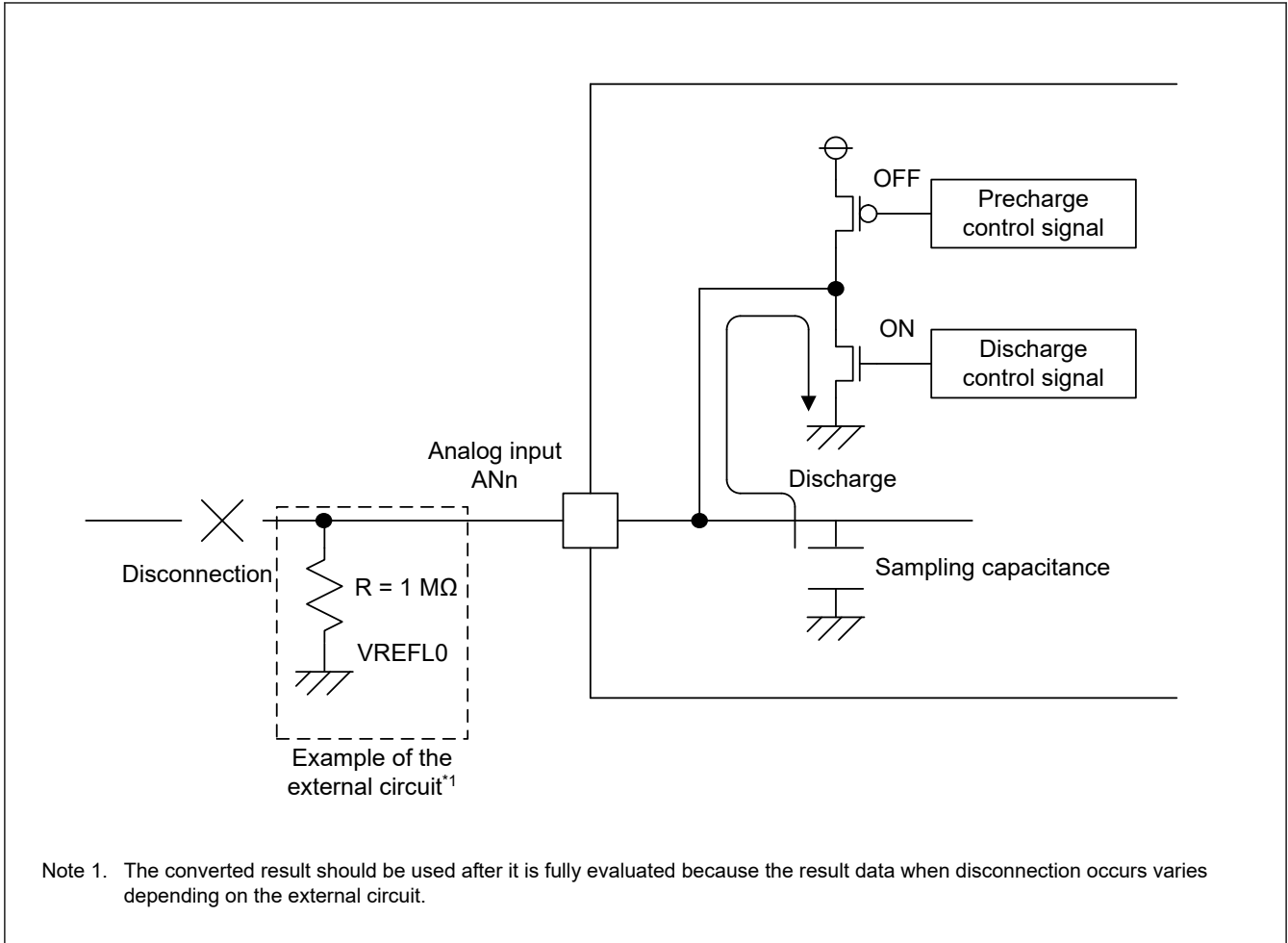


Figure 42.40 Example of disconnection detection when discharge is selected

### 42.3.10 Starting A/D Conversion with an Asynchronous Trigger

A/D conversion can be started by the input of an asynchronous trigger. To start A/D conversion by an asynchronous trigger, set the pin function in the  $PmnPFS$  register, set the A/D Conversion Start Trigger Select bits ( $ADSTRGR.TRSA[5:0]$ ) to  $0x00$ , then input a high-level signal to the asynchronous trigger ( $ADTRG_n$  ( $n = 0, 1$ ) pin). Finally, set both the  $ADCSR.TRGE$  and  $ADCSR.EXTRG$  bits to 1. Figure 42.41 shows timing of the asynchronous trigger input.

An asynchronous trigger cannot be selected in the A/D conversion start trigger for group B used in group scan mode. For details on setting the pin function, see section 19, I/O Ports.

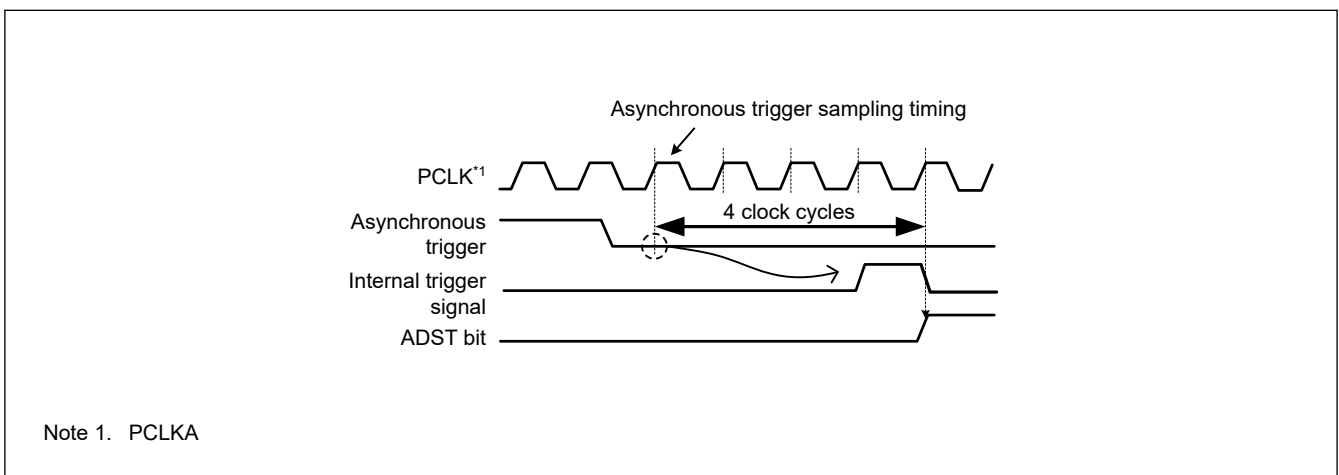


Figure 42.41 Asynchronous trigger input timing

### 42.3.11 Starting A/D Conversion with a Synchronous Trigger from a Peripheral Module

A/D conversion can be started by a synchronous trigger (GPT, ELC). To do this, set the ADCSR.TRGE bit to 1 and the ADCSR.EXTRG bit to 0, and select the relevant sources in the ADSTRGR.TRSA[5:0] bits and ADSTRGR.TRSA[5:0] bits.

### 42.3.12 Security Attribution between ADCn (n = 0, 1) and PORT (PmSAR (m = 0 to 9))

Only when the security attribution of ADC12n (n = 0, 1) match the security attribution of PORT (PmSAR), the correspond analog input channel can be set for ADC conversion. Otherwise the conversion result is unexpected.

Table 42.29 shows the condition of ADC for each Security Attribution setting.

**Table 42.29 Analog input pin conditions by security settings**

Security Attribution setting <sup>*1</sup>		ADC input channel select		Setting Combination result
ADC12n (n = 0, 1) security setting PSARD	PORT security setting PmSAR	ADANSA0 ADANSB0		
0 (Secure)	0 (Secure)	1 (channel correspond bit)		Input Valid
1 (Non-secure)	1 (Non-secure)	1 (channel correspond bit)		Input Valid
1 (Non-secure)	0 (Secure)	1 (channel correspond bit)		Input invalid and the conversion result is unexpected
0 (Secure)	1 (Non-secure)	1 (channel correspond bit)		Input invalid and the conversion result is unexpected

Note 1. VBATT 1/3 voltage monitor output do not have the Security Attribution judgement function.

### 42.3.13 Using Data Buffers

A ring buffer function consisting of 16 A/D data buffers is available. This function sequentially stores A/D conversion results other than self-diagnosis result (including addition/average results) in data buffers (ADBUF<sub>n</sub>, n = 0 to 15).

Each conversion result is stored at the timing when the A/D conversion result is stored in the data register, and most recent 16 conversion result data are retained.

The figure-below shows the schematic of data buffers, pointer, and overflow flag operations. When the BUFEN bit is set to 1, the A/D conversion result is transferred at each end of A/D conversion. The pointer indicates the number of data buffer to which the next transferred data is to be written. When data is written to up to buffer 15, the pointer is reset to 0000b and the overflow flag is set to 1. Subsequently transferred data overwrites the previously written data.

The overflow flag is reset to the initial value by writing 0x00 to the ADBUFPTR register.

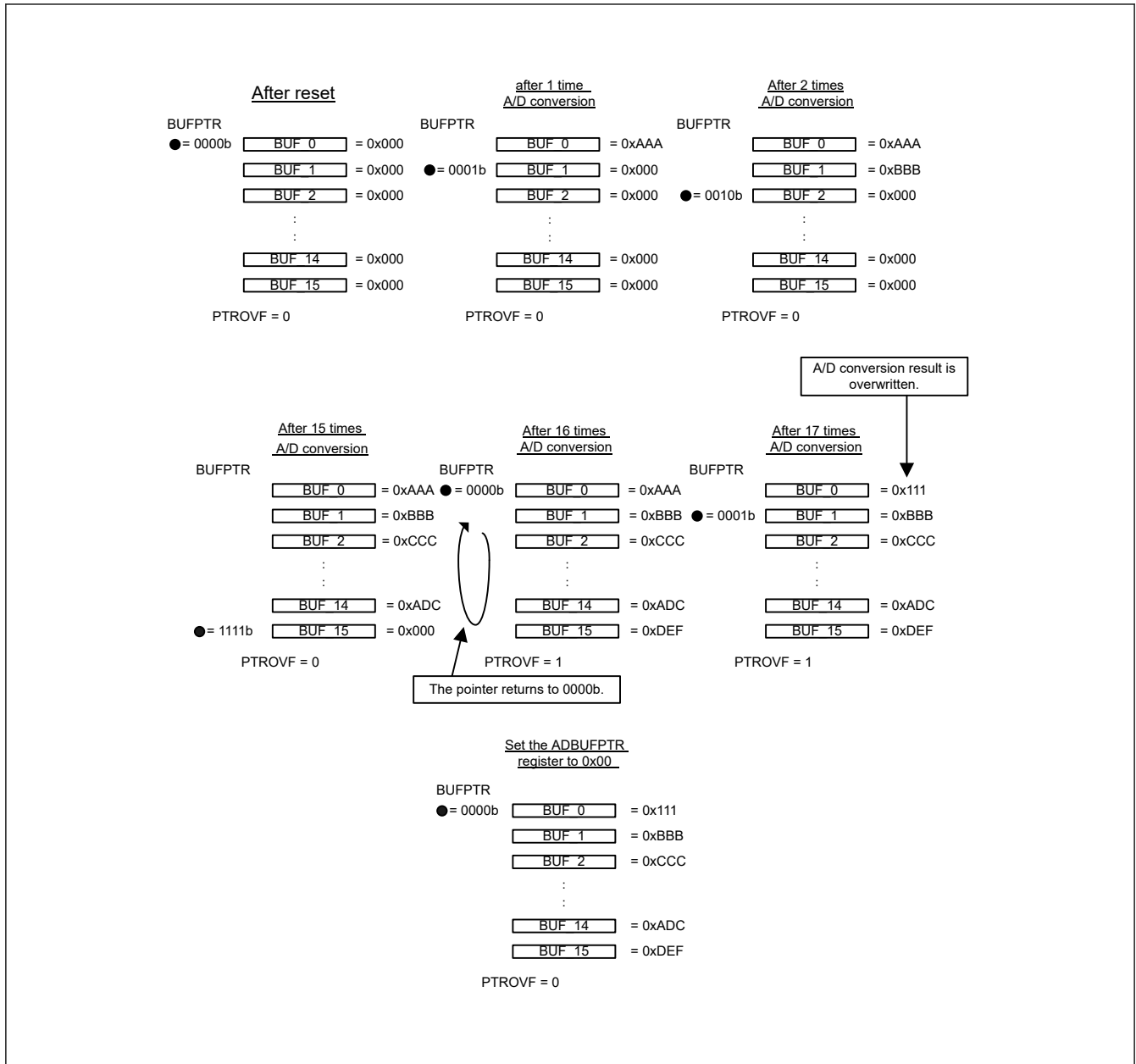


Figure 42.42 Data Buffers, Pointer, and Overflow Flag Operations

## 42.4 Interrupt Sources and DTC, DMAC Transfer Requests

### 42.4.1 Interrupt Requests

The ADC12 can send scan end interrupt requests ADC12i\_ADI (i = 0, 1) and ADC12i\_GBADI (i = 0, 1) to the CPU. The ADC12 also generates the ADC12i\_CMPAI (i = 0, 1)/ADC12i\_CMPBI (i = 0, 1) interrupt for the CPU in response to matches with a condition for comparison.

An ADC12i\_ADI (i = 0, 1) interrupt is always generated. An ADC12i\_GBADI (i = 0, 1) interrupt can be generated by setting the ADCSR.GBADIE bit to 1. Similarly, ADC12i\_CMPAI (i = 0, 1) and ADC12i\_CMPBI (i = 0, 1) interrupts can be generated by setting the ADCMPCR.CMPAIE and ADCMPCR.CMPBIE bit to 1.

In addition, the DTC or DMAC can be started when an ADC12i\_ADI (i = 0, 1) or an ADC12i\_GBADI (i = 0, 1) interrupt is generated. Using an ADC12i\_ADI (i = 0, 1) or ADC12i\_GBADI (i = 0, 1) interrupt to activate the DTC or DMAC to read the converted data enables continuous conversion without a burden on software.

Table 42.30 describes the interrupt sources and ELC events available for the ADC12.

Table 42.30 The interrupt source and ELC event of ADC12 (1 of 2)

Operation			Interrupt request or ELC event	Interrupt request	DTC or DMAC activation	ELC event request	Function
Scan mode	Double trigger mode	Compare function Window A/B					
Single scan mode	Deselected	Deselected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of single scan
		Selected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of single scan
			ADC12i_CMPAI (i = 0, 1)	✓	—	—	ADC12i_CMPAI (i = 0, 1) generated on a match comparison condition of Window A
			ADC12i_CMPBI (i = 0, 1)	✓	—	—	ADC12i_CMPBI (i = 0, 1) generated on a match comparison condition of Window B
			ADC12i_WCMPPM (i = 0, 1)	—	✓	✓	ADC12i_WCMPPM (i = 0, 1) generated on a match condition of the Window A/B compare function
			ADC12i_WCMPUM (i = 0, 1)	—	✓	✓	ADC12i_WCMPUM (i = 0, 1) generated on a mismatch condition of the Window A/B compare function
	Selected	Deselected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of scans in the even numbered times
Continuous scan mode	Deselected	Deselected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of scan of all selected channels
		Selected	ADC12i_CMPAI (i = 0, 1)	✓	—	—	ADC12i_CMPAI (i = 0, 1) generated on a match comparison condition of Window A
			ADC12i_CMPBI (i = 0, 1)	✓	—	—	ADC12i_CMPBI (i = 0, 1) generated on a match comparison condition of Window B

**Table 42.30** The interrupt source and ELC event of ADC12 (2 of 2)

Operation			Interrupt request or ELC event	Interrupt request	DTC or DMAC activation	ELC event request	Function
Scan mode	Double trigger mode	Compare function Window A/B					
Group scan mode	Deselected	Deselected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of group A scan
			ADC12i_GBADI (i = 0, 1)	✓	✓	—	ADC12i_GBADI (i = 0, 1) dedicated to group B generated at the end of group B scan
		Selected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of group A scan
			ADC12i_GBADI (i = 0, 1)	✓	✓	—	ADC12i_GBADI (i = 0, 1) dedicated to group B generated at the end of group B scan
			ADC12i_CMPAI (i = 0, 1)	✓	—	—	ADC12i_CMPAI (i = 0, 1) generated on a match comparison condition of Window A
			ADC12i_CMPBI (i = 0, 1)	✓	—	—	ADC12i_CMPBI (i = 0, 1) generated on a match comparison condition of Window B
	Selected	Deselected	ADC12i_ADI (i = 0, 1)	✓	✓	✓	ADC12i_ADI (i = 0, 1) generated at the end of group A scans in the even-numbered times
			ADC12i_GBADI (i = 0, 1)	✓	✓	—	ADC12i_GBADI (i = 0, 1) dedicated to group B generated at the end of group B scan

Note: ✓ available  
—: unavailable

For details on DTC settings, see [section 17, Data Transfer Controller \(DTC\)](#).

## 42.5 Event Link Function

### 42.5.1 Event Output to the ELC

The ELC uses the ADC12i\_ADI (i = 0, 1) interrupt request signal as an event signal ADC12i\_ADI (i = 0, 1), enabling link operation for the preset module. The ADC12i\_GBADI (i = 0, 1) interrupt and ADC12i\_CMPAI (i = 0, 1)/ADC12i\_CMPBI (i = 0, 1) interrupts cannot be used as an event signal. For details, see [Table 42.30](#).

An event signal can be output regardless of the settings of the corresponding interrupt request enable bits. For the scan end event (ADC12i\_ADI (i = 0, 1)), a high-level pulse for one PCLKA cycle is output at the same output timing as the interrupt output (ADC12i\_ADI (i = 0, 1)) shown in [Table 42.30](#). For a compare function match (ADC12i\_WCMPM (i = 0, 1)) and mismatch event (ADC12i\_WCMPUM (i = 0, 1)) to the ELC, a high-level pulse for one PCLKA cycle is output at the timing delayed by one cycle (PCLKA) from the interrupt output (ADC12i\_ADI (i = 0, 1)) shown in [Table 42.30](#).

To use compare function match (ADC12i\_WCMPM (i = 0, 1)) or mismatch event (ADC12i\_WCMPUM (i = 0, 1)) to the ELC, specify single-scan mode.

### 42.5.2 ADC12 Operation through an Event from the ELC

The ADC12 can start A/D conversion by the preset event specified in the ELSRn settings for the ELC as follows:

- Select the ELC\_AD00 (unit 0) signal in the ELC.ELSR8 register
- Select the ELC\_AD01 (unit 0) signal in the ELC.ELSR9 register



- Select the ELC\_AD10 (unit 1) signal in the ELC.ELSR10 register
- Select the ELC\_AD11 (unit 1) signal in the ELC.ELSR11 register.

If an ELC event occurs during A/D conversion, the event is disabled.

## 42.6 Usage Notes

### 42.6.1 Constraints on Setting the Registers

Set each register while the ADCSR.ADST bit is 0.

### 42.6.2 Constraints on Reading the Data Registers

The following registers must be read in halfword units:

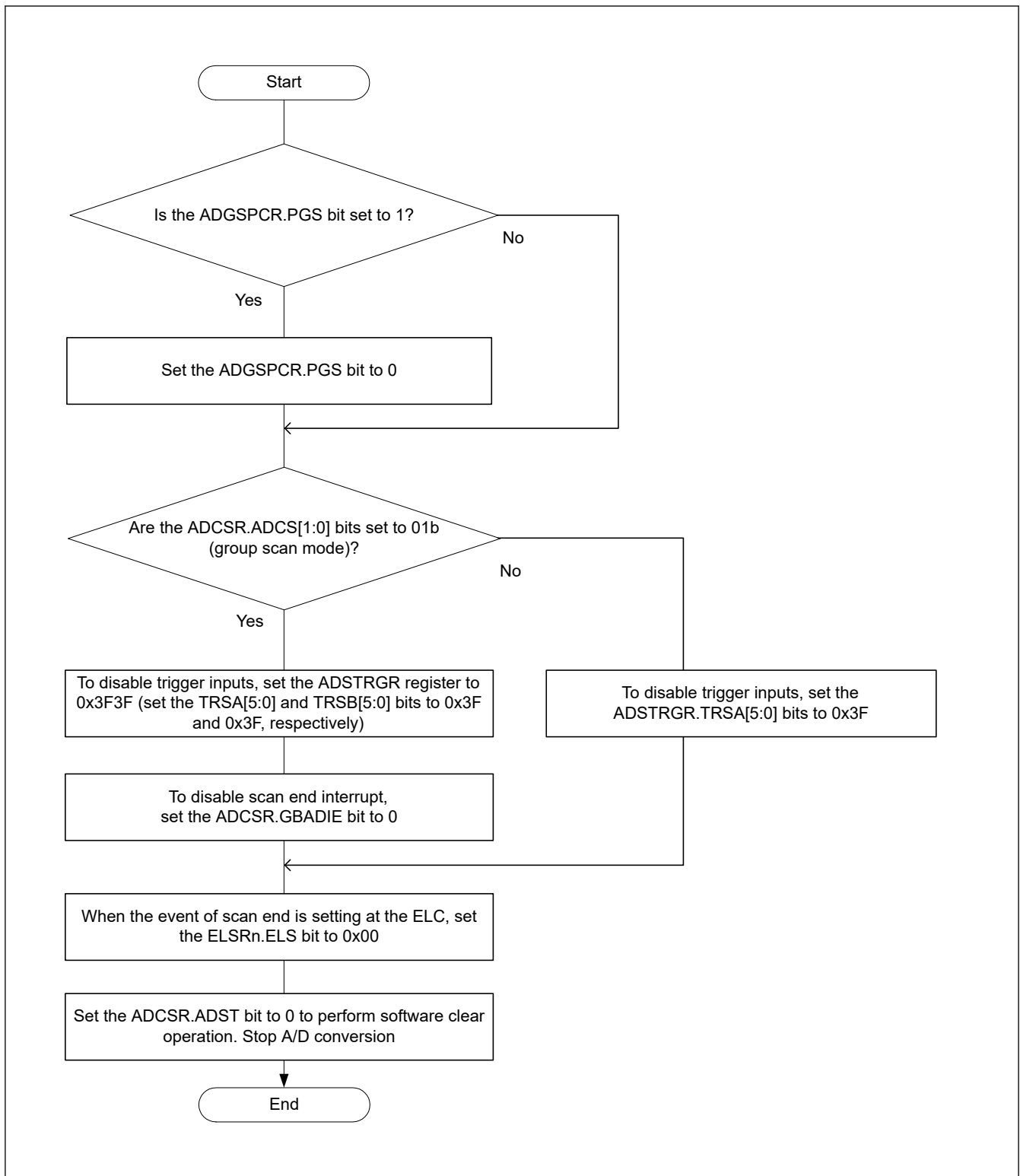
- A/D Data Registers
- A/D Data Duplexing Register
- A/D Data Duplexing Register A
- A/D Data Duplexing Register B
- A/D Temperature Sensor Data Register
- A/D Internal Reference Voltage Register
- A/D VBATT monitor Data Register
- A/D Self-Diagnosis Data Register
- A/D Data Buffer Registers n (N = 0 to 15)

If a register is read twice in byte units, that is, the upper byte and lower byte are read separately, the A/D-converted value read initially might disagree with the A/D-converted value read subsequently. To prevent this, never read the data registers in byte units.

### 42.6.3 Constraints on Stopping A/D Conversion

#### (1) A/D Conversion Stop Procedure

To stop A/D conversion when an asynchronous trigger or a synchronous trigger is selected as the condition for starting A/D conversion, follow the procedure shown in [Figure 42.43](#).



**Figure 42.43 Procedures for clearing the ADCSR.ADST bit by software**

To specify the following settings after performing the clear operation by software, provide a wait period for at least two ADCLK cycles.

- Enabling scan end interrupts
- Enabling scan end events for the event link controller
- Starting A/D conversion by software
- Enabling trigger input

## (2) Notes on Modes and Status Bits

If necessary, individually initialize or set again the voltage status for self-diagnosis, the judgment of the even number or odd number specified for double-trigger mode, and the monitor flags of the compare function.

- To set again the voltage status for self-diagnosis, set the ADCER.DIAGLD bit to 1 and then set a desired value in the ADCER.DIAGVAL[1:0] bits.
- If the setting of the ADCSR.DBLE bit is changed from 0 to 1, the double-trigger mode operation starts from the first scanning.
- To initialize the monitor flags of the compare function (MONCMPA, MONCMPB, and MONCOMB), set the ADCMPCR.CMPAE and ADCMPCR.CMPBE bits to 0.
- To initialize the data storage buffer pointer, set the ADBUFPTR register to 0x00.
- To initialize the continuous sampling function (ADSHMSR.SHMD = 1), set ADCSHMSR.SHMD to 0. To use the continuous sampling function again after initialization, set ADCSHMSR.SHMD to 1 after waiting for 1 cycle of ADCLK or longer.

### 42.6.4 A/D Conversion Restart and Termination Timing

A maximum of 6 ADCLK cycles is required for the idle analog unit of the ADC12 to restart on setting the ADCSR.ADST bit to 1. A maximum of 2 ADCLK cycles is required for the operating analog unit of the ADC12 to terminate on setting the ADCSR.ADST bit to 0. A maximum of 1 PCLKA + 1 ADCLK cycles is required for the operating analog unit of the ADC12 to terminate on setting the ADCSR.ADST bit to 0 when PCLKA to ADCLK frequency ratio is 1:2 or 1:4.

### 42.6.5 Constraints on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data. This occurs when the CPU does not complete the reading of the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated.

### 42.6.6 Settings for the Module-Stop Function

The Module Stop Control Register can enable or disable ADC12 operation. The ADC12 is initially stopped after a reset. The registers become accessible on release from the module-stop state. After release from the module-stop state, wait for at least 1  $\mu$ s before starting A/D conversion. For details, see [section 10, Low Power Modes](#).

### 42.6.7 Notes on Entering the Low-Power States

Before entering the module-stop state or Software Standby mode, be sure to stop A/D conversion. Set the ADCSR.ADST bit to 0 and secure certain period until the analog unit of the ADC12 stops. Follow the procedure shown in [Figure 42.43](#) to clear the ADCSR.ADST bit with software. Then, wait for 2 clock cycles of ADCLK before entering the module-stop state or Software Standby mode.

### 42.6.8 Notes on Canceling Software Standby Mode

After software standby mode is canceled, wait at least 1  $\mu$ s after the stabilization time for the oscillator elapses and before starting A/D conversion. For details, see [section 10, Low Power Modes](#)

### 42.6.9 Error in Absolute Accuracy When Disconnection Detection Assistance Is in Use

Using disconnection detection assistance leads to an error in absolute accuracy of the ADC12. This error arises because an erroneous voltage is input to the analog input pins due to the resistive voltage division between the pull-up or pull-down resistor ( $R_p$ ) and the resistance of the signal source ( $R_s$ ). This error in absolute accuracy is calculated from the following formula:

$$\text{Maximum error in absolute accuracy (LSB)} = (2^{\text{Resolution}} - 1) \times R_s / (R_s + R_p)$$

Only use disconnection detection assistance after thorough evaluation.

### 42.6.10 Constraints on Operating Modes and Status Bits

Initialize or set again individually, if necessary, the voltage values in self-diagnosis, the value of the first scan or second scan in double trigger mode, the data buffer pointer, and status monitor in the compare function.

- Select the voltage values in self-diagnosis (ADCER.DIAGVAL[1:0]) after setting ADCER.DIAGLD to 1.
- Double-trigger mode operates as the first scan after setting ADCSR.DBLE from 0 to 1.
- The status monitor bits (MONCMPA, MONCMPB, MONCOMB) in the compare function are initialized after setting ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0.

### 42.6.11 Notes on Board Design

The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or be placed near each other. If these rules are not followed, noise can occur on analog signals and A/D conversion accuracy is affected. The analog input pins, reference power supply pin (VREFH0, VREFH), reference ground pin (VREFL0, VREFL), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0). The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

### 42.6.12 Constraints on Noise Prevention

To prevent the analog input pins from being destroyed by abnormal voltage such as excessive surge, insert a capacitor between AVCC0 and AVSS0 and between VREFH0 and VREFL0 and between VREFH and VREFL. Additionally, connect a protection circuit to protect the analog input pins as shown in [Figure 42.44](#).

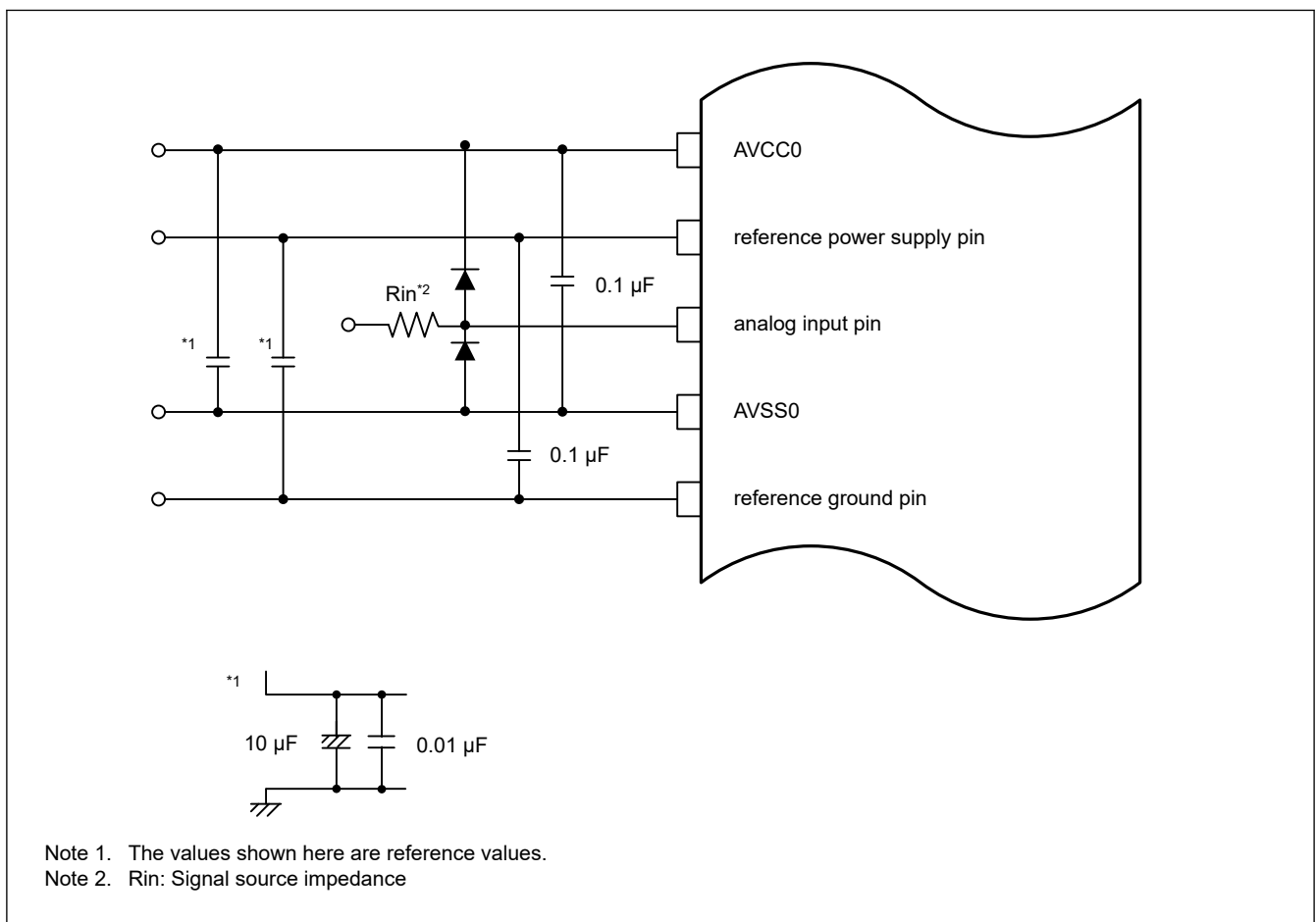


Figure 42.44 Example protection circuit for analog inputs

### 42.6.13 Port Settings When Using the ADC12 Input

When using the high-precision channels, do not use PORT0 as digital output ports. Renesas recommends that you do not use the digital output that is also used as the AD analog input if normal-precision channel is used. If the digital output that is also used as the AD analog input is used for output signals, perform A/D conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results.

### 42.6.14 Relationship between ADC12 Units 0 and 1 and the ACMPHS

For the A/D conversion targets in [Table 42.31](#), unit 0 and 1 cannot perform A/D conversion at the same time.

**Table 42.31 A/D conversion targets that are mutually exclusive with each other**

A/D conversion target	
Unit 0	Unit 1
Temperature sensor	
Internal reference voltage	
VBATT 1/3 voltage monitor output	

The A/D conversion targets in [Table 42.32](#) should not be selected as ACMPHS input during A/D conversion, because these pins are multiplexed with the ADC12 and ACMPHS.

**Table 42.32 A/D conversion targets that are mutually exclusive with ACMPHS**

A/D conversion target		
Unit 0	Unit 1	ACMPHS
AN000	-	ACMPHS0.IVCMP2
AN002	-	ACMPHS0.IVCMP3
AN005	-	ACMPHS0.IVCMP0
AN007/DA0	-	ACMPHS0.IVREF3
Internal reference voltage	-	ACMPHS0.IVREF2
-	AN100	ACMPHS1.IVCMP2
-	AN101	ACMPHS1.IVREF0
-	AN102	ACMPHS1.IVCMP3
-	AN104	ACMPHS1.IVREF1
-	Internal reference voltage	ACMPHS1.IVREF2

### 42.6.15 Calculation for Sampling Time

The sampling time can be easily estimated by the following figure and formula. This is the time to reach the voltage within 1/4 LSB.

$$t_{SPL} = (R_{EXT} + R_{AD}) \times (C_{EXT} + C_{AD}) \times \ln(C_{AD} / (C_{EXT} + C_{AD}) \times 2^{N+2})$$

$R_{EXT}$  shows external signal source impedance

$C_{EXT}$  shows external capacitance (pin capacitance<sup>\*1</sup> + PCB parasitic capacitance)

$N = 12, 10$  or  $8$  (conversion resolution)

$C_{AD} = 5$  pF (internal capacitance)

$R_{AD} = 1.0$  k $\Omega$  (internal resistance, case of high-speed channels)

$R_{AD} = 2.0$  k $\Omega$  (internal resistance, case of normal-speed channels)

Note 1. Typical value of analog input pin is 5 pF

For example, if  $R_{EXT}$  is 1 k $\Omega$ ,  $C_{EXT}$  is 10 pF and  $N$  is 12 bits,  $t_{SPL}$  of high-speed channel is 258 ns.

This formula simplifies the general use case. This formula is not guaranteed and should be used only for estimation.

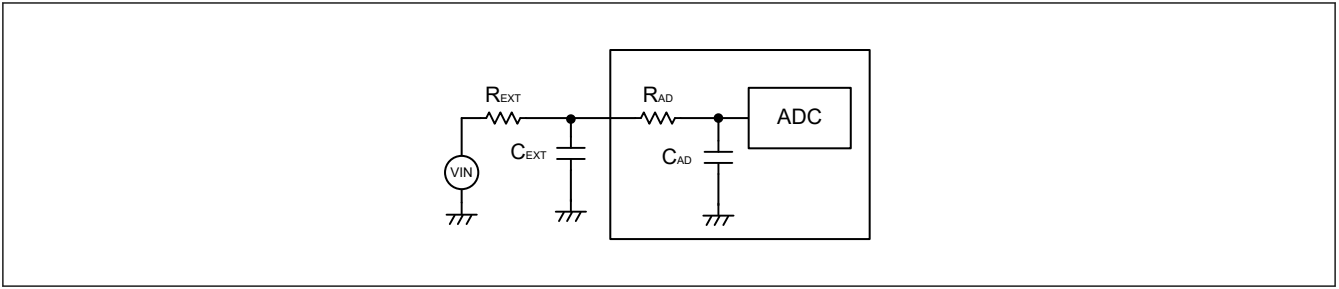


Figure 42.45 Sample and hold circuit simplified diagram

#### 42.6.16 Notes on simultaneous operation with other modules

The following combinations may degrade the ADC accuracy depending on the specific timing of operation.

1. AN000 without channel-dedicated sample-and-hold and AN104 mutually.
2. AN007 and AN105 mutually.
3. AN000, AN001 using channel-dedicated sample-and-hold when ADC unit 1, DAC12 or ACMPHS is operated.
4. High-precision channels of ADC unit1 when AN000, AN001 using channel-dedicated sample-and-hold are operated.
5. All channels of ADC unit 1 when ACMPHS is operated.
6. All channels of ADC unit 1 when DAC12 is operated.

To mitigate deterioration, it is recommended to take following measures.

- Performing A/D conversion with average mode.
- For even better results, performing A/D conversion several times, eliminate the maximum and minimum values, and obtain the average of the other results. (For the condition No.3, only performing average mode may not be effective depending on the noise. Eliminating the maximum and minimum is recommended.)
- For the condition No.6, using DA AD synchronous conversion set by DAADSCR.DAADST and DAADUSR.AMADESEL1 bits of DAC12.

Depending on the environment, ADC accuracy may be deteriorated even under conditions not listed here.

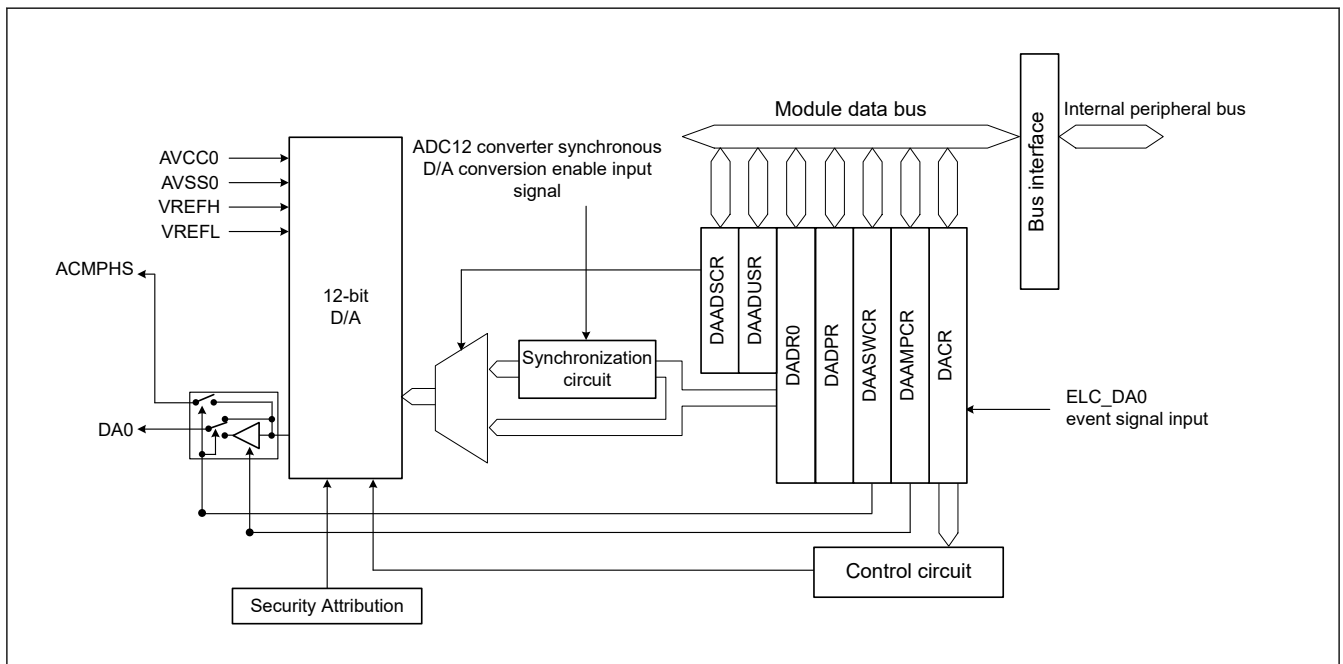
## 43. 12-Bit D/A Converter (DAC12)

### 43.1 Overview

The MCU provides a 12-bit D/A Converter (DAC12) with an output amplifier. [Table 43.1](#) lists the DAC12 specifications, [Figure 43.1](#) shows a block diagram, and [Table 43.2](#) lists the I/O pins.

**Table 43.1 DAC12 specifications**

Parameter	Specifications
Resolution	12 bits
Output channels	1 channel
Interference reduction between analog modules	Methods provided to minimize interference between D/A and A/D conversion: <ul style="list-style-type: none"> <li>• D/A converted data update timing is controlled by the synchronous D/A conversion enable input signal from the ADC12 (unit 1)</li> <li>• Degradation of A/D conversion accuracy caused by interference is reduced by controlling the DAC12 inrush current generation timing with the enable signal.</li> </ul>
Module-stop function	Module-stop state can be set to reduce power consumption
Event link function (input)	The DA0 conversion can be started on input of an event signal
D/A output amplifier control function	Controls whether the output amplifier (for both amplifier-through and amplifier-bias controls) is used
Destination of D/A output control function	Controls whether the output to the external pin or to the internal modules (ACMPHS) is used
TrustZone Filter	Security and Privilege attribution can be set



**Figure 43.1 DAC12 block diagram**

[Table 43.2](#) lists the pin configuration of the DAC12.

**Table 43.2 DAC12 I/O pins (1 of 2)**

Pin name	I/O	Function
AVCC0	Input	<ul style="list-style-type: none"> <li>• Analog power and analog reference top voltage supply pin for ADC12 and DAC12.</li> <li>• Connect to VCC when these modules are not used.</li> </ul>
AVSS0	Input	<ul style="list-style-type: none"> <li>• Analog ground and analog reference ground supply pin for ADC12 and DAC12.</li> <li>• Connect to VSS when these modules are not used.</li> </ul>
VREFH	Input	Analog reference top voltage supply pin for the ADC12 (unit 1) and the DAC12

**Table 43.2 DAC12 I/O pins (2 of 2)**

Pin name	I/O	Function
VREFL	Input	Analog reference ground pin for the ADC12 (unit 1) and the DAC12
DA0	Output	Channel 0 output pin for the analog signals processed by the DAC12

## 43.2 Register Descriptions

### 43.2.1 DADRn : D/A Data Register n (n = 0)

Base address: DAC12 = 0x4033\_3000  
 DAC12\_NS = 0x5033\_3000

Offset address: 0x00 + 0x02 × n

Bit position: 15 0

Bit field:



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Note: S-TYPE-3, P-TYPE-3

DADRn register is 16-bit read/write registers that store data for D/A conversion. When an analog output is enabled, the values in DADRn are converted and output to the analog output pins.

12-bit data can be formatted as left- or right-justified in the DADPR.DPSEL bit setting. In right-justified format (DADPR.DPSEL = 0), the lower 12 bits, [11:0], are valid. In left-justified format (DADPR.DPSEL = 1), the upper 12 bits, [15:4], are valid.

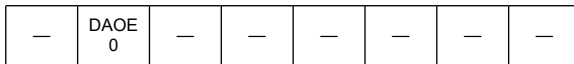
### 43.2.2 DACR : D/A Control Register

Base address: DAC12 = 0x4033\_3000  
 DAC12\_NS = 0x5033\_3000

Offset address: 0x04

Bit position: 7 6 5 4 3 2 1 0

Bit field:



Value after reset: 0 0 0 1 1 1 1 1

Bit	Symbol	Function	R/W
4:0	—	These bits are read as 1. The write value should be 1.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	DAOE0	D/A Output Enable 0 0: Disable analog output of channel 0 (DA0) 1: Enable D/A conversion of channel 0 (DA0)	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Only set this register while the ADC12 is halted when the DAADSCR.DAADST bit is 1 (interference reduction between D/A and A/D conversion is enabled). Only set DACR while the ADCSR.ADST bit is 0 and after selecting the software trigger, for the ADC12 trigger to securely stop the ADC12. This MCU only supports ADC12 (unit 1).

#### DAOE0 bit (D/A Output Enable 0)

The DAOE0 bit controls D/A conversion, amplifier operation, and analog output in combination with the DAAMPCR.DAAMP0 bit and the DAASWCR.DAASW0 bit. See [Table 43.3](#).

When the DAOE0 bit is 0, D/A conversion is not processed, and no conversion result is output.



When interference reduction between D/A and A/D conversions is enabled (DAADSCR.DAADST = 1), set the DAOE0 bit while the ADCSR.ADST bit of the ADC12 (unit 1) is set to 0. Then, select the software trigger for the ADC12 (unit 1) trigger to securely stop the ADC12 (unit 1).

The event link function can be used to set the DAOE0 bit to 1. The DAOE0 bit is set to 1 when the event specified in the ELSR12 register of the ELC (ELC\_DA0 event) occurs, and output of the D/A conversion results starts.

**Table 43.3 D/A conversion and analog output control**

DACR	DAAMPCR	DAASWCR	Operation	Amplifier operation	Analog external output*1	Analog internal output*2
DAOE0	DAAMP0	DAASW0				
0	x	x	Stop	Stop	Hi-Z	Hi-Z
1	0	0	Run	Stop	Amplifier-through	Hi-Z
1	0	1	Run	Stop	Hi-Z	Amplifier-through
1	1	0	Run	Run	Amplifier output	Hi-Z
1	1	1	Run	Run	Hi-Z	Hi-Z

Note: x : Don't care

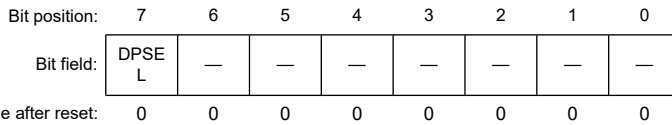
Note 1. output to pin

Note 2. output to ACMPHS

### 43.2.3 DADPR : DADRn Format Select Register

Base address: DAC12 = 0x4033\_3000  
 DAC12\_NS = 0x5033\_3000

Offset address: 0x05



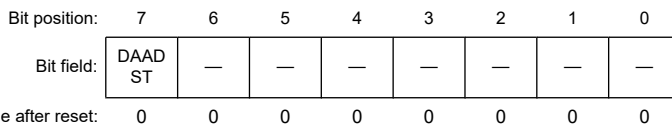
Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	DPSEL	DADRn Format Select 0: Right-justified format 1: Left-justified format	R/W

Note: S-TYPE-3, P-TYPE-3

### 43.2.4 DAADSCR : D/A A/D Synchronous Start Control Register

Base address: DAC12 = 0x4033\_3000  
 DAC12\_NS = 0x5033\_3000

Offset address: 0x06



Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	DAADST	D/A A/D Synchronous Conversion 0: Do not synchronize DAC12 with ADC12 (unit 1) operation (disable interference reduction between D/A and A/D conversion). 1: Synchronize DAC12 with ADC12 (unit 1) operation (enable interference reduction between D/A and A/D conversion).	R/W

Note: S-TYPE-3, P-TYPE-3

To minimize interference between D/A and A/D conversion, the DAADSCR register enables synchronization of the start timing of D/A conversion with the ADC12 synchronous D/A conversion enable input signal.

Only set this register while the ADC12 (unit 1) is halted, that is, while the ADCSR.ADST bit is 0 after selecting the software trigger as the ADC12 (unit 1) trigger.

Select unit 1 as the target ADC12 unit before setting the DAADST bit to 1. Set DAADUSR.AMADESEL1 bit to 1 to select unit 1. This MCU only supports ADC12 unit 1.

**DAADST bit (D/A A/D Synchronous Conversion)**

Setting the DAADST bit to 0 allows the DADRn register value to be converted into analog data at any time. Setting the DAADST bit to 1 allows synchronization of D/A conversion with the synchronous D/A conversion enable input signal from the ADC12 (unit 1). With this bit set, D/A conversion does not start until the ADC12 (unit 1) completes A/D conversion, even when the DADRn register is changed.

Set this bit while the ADCSR.ADST bit is set to 0. Then, select the software trigger for the ADC12 (unit 1) trigger to securely stop the ADC12 (unit 1). Set the DAADUSR.AMADESEL1 bit to 1 before setting the DAADST bit to 1.

The event link function cannot be used when the DAADST bit is set to 1. Stop the event link function by setting the ELSR12 register of the ELC.

**43.2.5 DAAMPCR : D/A Output Amplifier Control Register**

Base address: DAC12 = 0x4033\_3000  
 DAC12\_NS = 0x5033\_3000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	DAAMP0	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	DAAMP0	Amplifier Control 0 0: Do not use channel 0 output amplifier 1: Use channel 0 output amplifier	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The DAAMPCR register selects D/A output with or without using the amplifier.

**DAAMP0 bit (Amplifier Control 0)**

When the DAAMP0 bit is 0, analog values are output for D/A output of channel 0 without using the amplifier. When the DAAMP0 bit is 1, analog values are output for D/A output of channel 0 through the amplifier.

When the DACR.DAOE0 bit is 0, the amplifier is not used regardless of the setting of the DAAMP0 bit. See Table 43.3 for details.

**43.2.6 DAASWCR : D/A Amplifier Stabilization Wait Control Register**

Base address: DAC12 = 0x4033\_3000  
 DAC12\_NS = 0x5033\_3000

Offset address: 0x1C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	DAASW0	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	DAASW0	D/A Amplifier Stabilization Wait 0 and D/A internal output control 0: For output to external pin: Amplifier stabilization wait off (output) for channel 0 For output to internal module: Disable output for channel 0 1: For output to external pin: Amplifier stabilization wait on (high-Z) for channel 0 For output to internal module: Enable output for channel 0	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The DAASWCR register controls D/A output with the output amplifier or D/A output for internal modules. This register is used in the initialization procedure to wait for stabilization of the D/A output amplifier. DAASW0 bit in DAASWCR should be set to 1 when the DACR.DAOE0 bit is 0. See [section 43.7.5. Initialization Procedure with the Output Amplifier](#).

#### DAASW0 bit (D/A Amplifier Stabilization Wait 0)

Set the DAASW0 bit to 1 in the initialization procedure to wait for the stabilization of the D/A channel 0 output amplifier. When DAASW0 is set to 1, D/A conversion operates, but the conversion result of D/A is not output from channel 0 to the DA0 pin. When the DAASW0 bit is 0, the stabilization wait time stops, and the D/A conversion result of channel 0 is output through the output amplifier to the DA0 pin. When the amplifier is not used (DAAMPCR.DAAMP0 bit is 0) and DAASW0 is set to 1, D/A conversion result of channel 0 is output to the internal modules.

### 43.2.7 DAADUSR : D/A A/D Synchronous Unit Select Register

Base address: DAC12 = 0x4033\_3000  
 DAC12\_NS = 0x5033\_3000

Offset address: 0x10C0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	AMADSEL1	—

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	AMADSEL1	A/D Unit 1 Select 0: Do not select unit 1 1: Select unit 1	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

The DAADUSR register selects the target ADC12 unit for D/A and A/D synchronous conversions. Set the AMADSEL1 bit to 1 to select unit 1 as the target synchronous unit for the MCU. When setting the DAADSCR.DAADST bit to 1 for synchronous conversions, select the target unit in this register in advance.

Only set the DAADUSR register while the ADCSR.ADST bit of the ADC12 is set to 0 and the DAADSCR.DAADST bit is set to 0.

## 43.3 Operation

When the DAOE0 bit in DACR is set to 1, DAC12 is enabled and the conversion result is output.

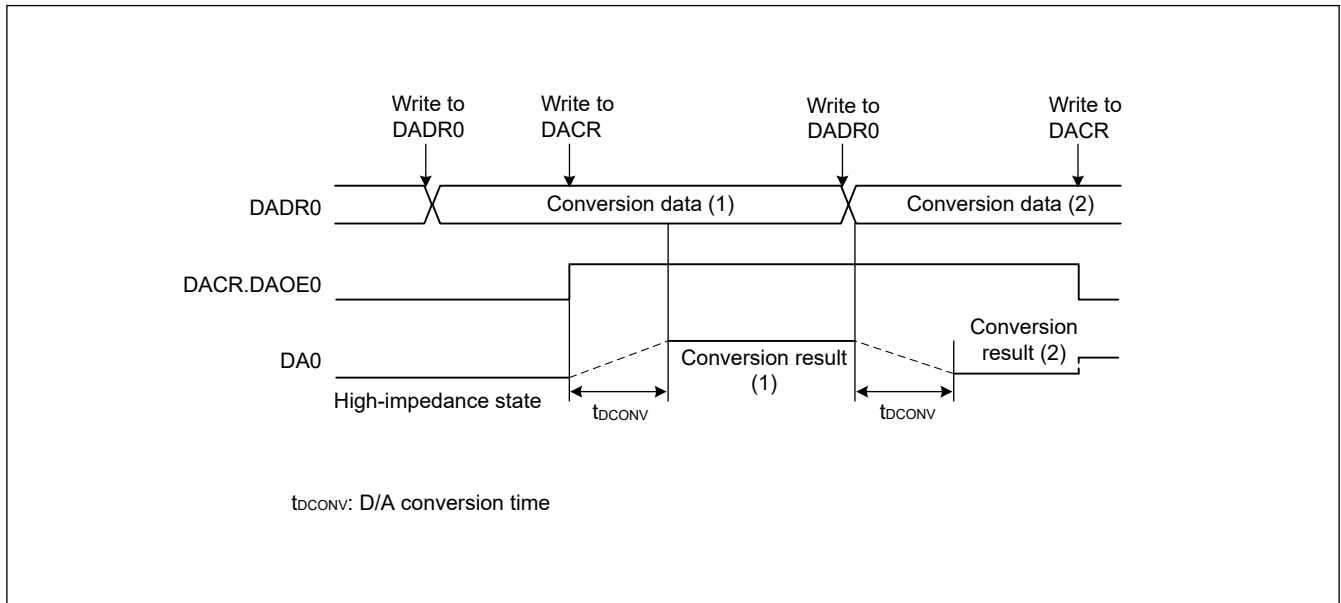
This following example shows D/A conversion. [Figure 43.2](#) shows the timing of this operation.

To process D/A conversion:

1. Set the data for D/A conversion in the DADR0 register and the data format in the DADPR.DPSEL bit.
2. Set the DACR.DAOE0 bit to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time  $t_{DCONV}$  elapses. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is set to 0. The output value (reference) is expressed by the following formula:

$$\frac{\text{Setting in DADR0}}{4096} \times VREFH$$

- To start conversion again, write another value to DADR0. The conversion result is output after the conversion time  $t_{\text{D CONV}}$  elapses.  
When the DAADSCR.DAADST bit is 1 (interference reduction between D/A and A/D conversion is enabled), a maximum of one A/D conversion time is required for D/A conversion to start. When ADCLK is faster than the peripheral clock, a longer time might be required.
- To disable analog output, set the DAOE0 bit to 0.



**Figure 43.2 Example of DAC12 operation**

### 43.3.1 Reducing Interference between D/A and A/D Conversion

When D/A conversion starts, the DAC12 generates inrush current. Because the DAC12 and ADC12 (unit 1) share the same analog power supply, the generated inrush current can interfere with ADC12 (unit 1) operation.

While the DAADSCR.DAADST bit is 1, D/A conversion does not start immediately on updating the DADR<sub>m</sub> register. Instead:

- If the DADR<sub>m</sub> register data is modified while the ADC12 is halted, D/A conversion starts in 1 PCLKA cycle.
- If the DADR<sub>m</sub> register data is modified while the ADC12 is performing a 12-bit A/D conversion, D/A conversion starts on A/D conversion completion. Therefore, it takes up to one A/D conversion time period for the DADR<sub>m</sub> register data update to be reflected as the D/A conversion circuit output. Until the D/A conversion completes, the DADR<sub>m</sub> register value does not correspond to the analog output value.

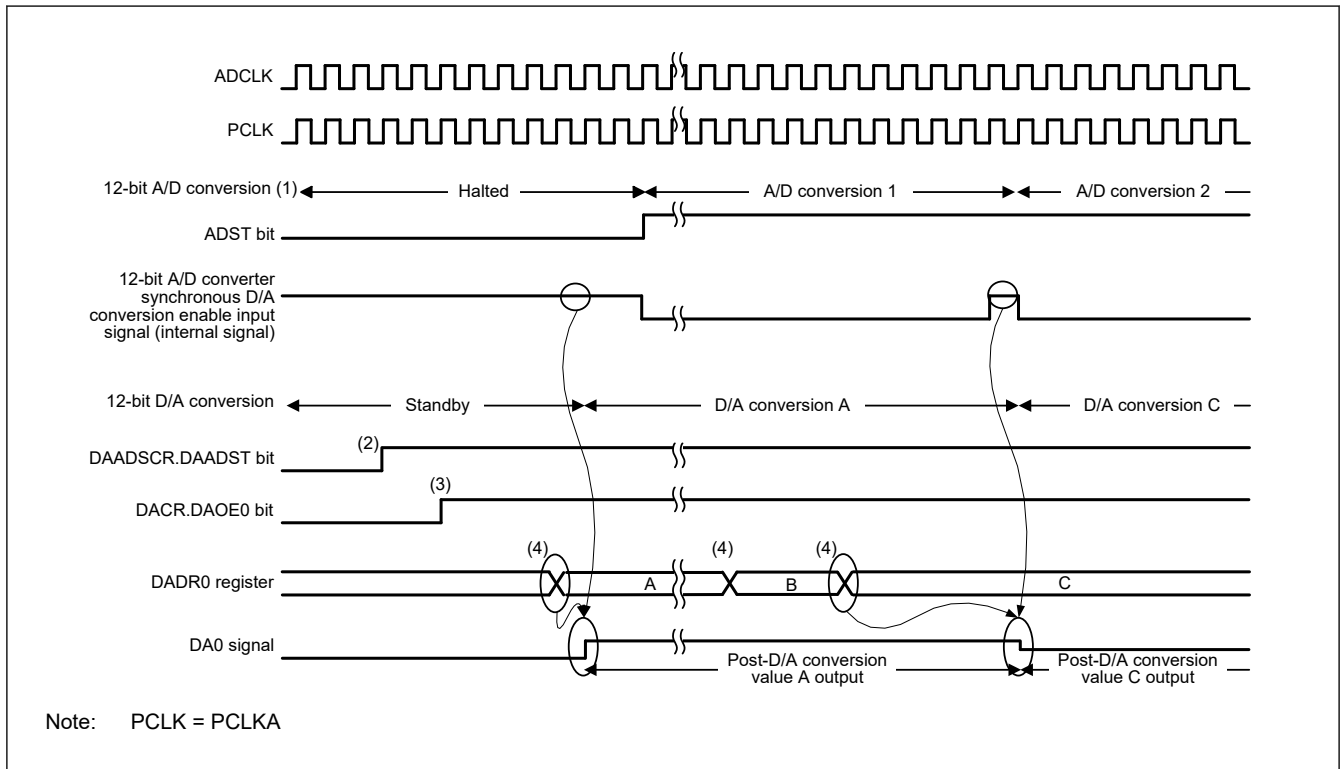
When the DAADSCR.DAADST bit is 1, it is not possible to check through software whether the DADR<sub>m</sub> register value was D/A-converted.

The following sequence provides an example of channel 0 D/A conversion, in which the DAC12 is synchronized with the ADC12 (unit 1). [Figure 43.3](#) shows the timing of this operation.

To perform D/A conversion on channel 0 in synchronization with the ADC12 (unit 1):

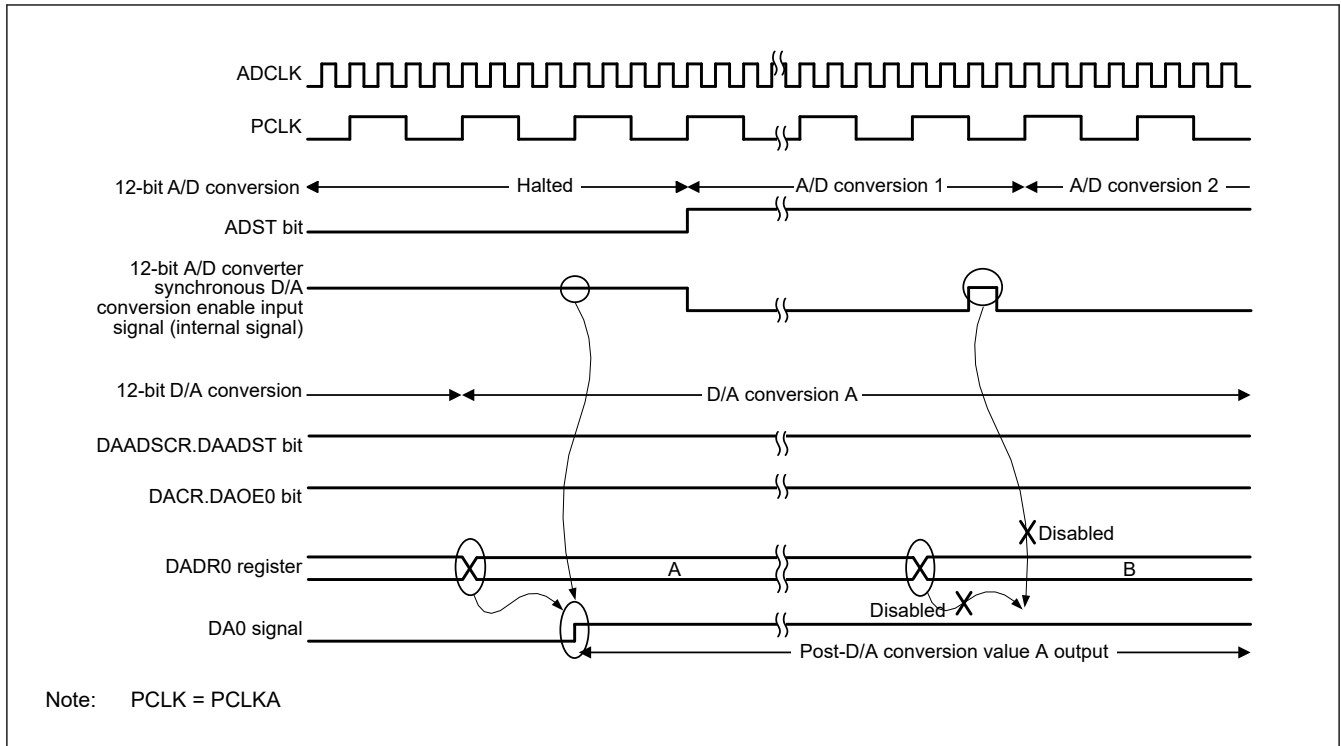
- Confirm that the ADC12 (unit 1) is halted and set the DAADUSR.AMA DSEL1 bit to 1.
- Confirm that the ADC12 (unit 1) is halted and set the DAADSCR.DAADST bit to 1.
- Confirm that the ADC12 (unit 1) is halted and set the DACR.DAOE0 bit to 1.
- Set the DADR0 register. If ADCLK is faster than the peripheral clock, D/A conversion might be delayed for longer than one A/D conversion time.

- If the ADC12 (unit 1) is halted (ADCSR.ADST = 0) when the DADR0 register is modified, D/A conversion starts in 1 PCLKA cycle.
- If the 12-bit A/D conversion is in progress (ADCSR.ADST = 1) when the DADR0 register is modified, D/A conversion starts on A/D conversion completion. If the DADR0 register is modified twice during A/D conversion, the first update might not be converted.



**Figure 43.3 Example conversion when DAC12 is synchronized with ADC12 (unit 1)**

When ADCLK is faster than PCLKA, the DAC12 might not be able to capture the synchronous D/A conversion enable input signal from the ADC12 (unit 1) during the 1 ADCLK cycle that is output between A/D conversion 1 and A/D conversion 2, as shown in Figure 43.4. In this case, post-D/A conversion value A is continuously output as the DA0 signal.



**Figure 43.4** Example when the DAC12 cannot capture the synchronous D/A conversion enable input signal from the ADC12 (unit 1)

### 43.4 Event Link Operation Setting Procedure

This section describes the procedures used in event link operation.

#### 43.4.1 DA0 Event Link Operation Setting Procedure

To set up DA0 event link operation:

1. Set the DADPR.DPSEL bit and the data for D/A conversion in the DADR0 register.
2. Set the ELC\_DA0 event signal to be linked to each peripheral module in the ELSR12 register.
3. Set the ELCR.ELCON bit to 1. This enables event link operation for all modules with the event link function selected.
4. Set the event output source module to activate the event link. After the event is output from the module, the DACR.DAOE0 bit becomes 1, and D/A conversion starts on channel 0.
5. Set the ELSR12 register to 0x0000 to stop event link operation of DAC12 channel 0. All event link operation is stopped when the ELCR.ELCON bit is set to 0.

### 43.5 Analog Output Control by Security Attribution

Analog output to DA0 pin is controlled by the security attribution of Port and DAC12 as shown in [Table 43.4](#).

Analog output to ACPHPS is not controlled by the security attribution.

**Table 43.4** Analog output control by security attribution (1 of 2)

Security Attribution		Analog Output
DAC12 (PSARD)	P014 (PmSAR (m = 0 to 9))	P014 pin output
0 (secure)	1 (non-secure)	enable
0 (secure)	0 (secure)	enable
1 (non-secure)	1 (non-secure)	enable
1 (non-secure)	0 (secure)	disable

**Table 43.4 Analog output control by security attribution (2 of 2)**

Security Attribution		Analog Output
DAC12 (PSARD)	P014 (PmSAR (m = 0 to 9))	P014 pin output
1 (non-secure)	0 (secure)	disable

### 43.6 Usage Notes on Event Link Operation

- When the event link function is used, do not use the amplifier output function.
- When the event specified for the ELC\_DA0 event signal is generated while a write to the DACR.DAOE0 bit is performed, the write cycle is stopped, and the generated event takes precedence in setting the bit to 1.
- Use of the event link function is prohibited when the DAADSCR.DAADST bit is set to 1 to reduce interference between D/A and A/D conversions.

### 43.7 Usage Notes

#### 43.7.1 Settings for the Module-Stop Function

DAC12 operation can be disabled or enabled using the Module Stop Control Register. The DAC12 is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

#### 43.7.2 DAC12 Operation in the Module-Stop State

When the MCU enters the module-stop state with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in the module-stop state, disable D/A conversion by setting the DACR.DAOE0 bit to 0.

#### 43.7.3 DAC12 Operation in Software Standby Mode

When the MCU enters Software Standby mode with D/A conversion enabled, the D/A outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current must be reduced in Software Standby mode, disable D/A conversion by setting the DACR.DAOE0 bit to 0.

#### 43.7.4 Constraint on Entering Deep Software Standby Mode

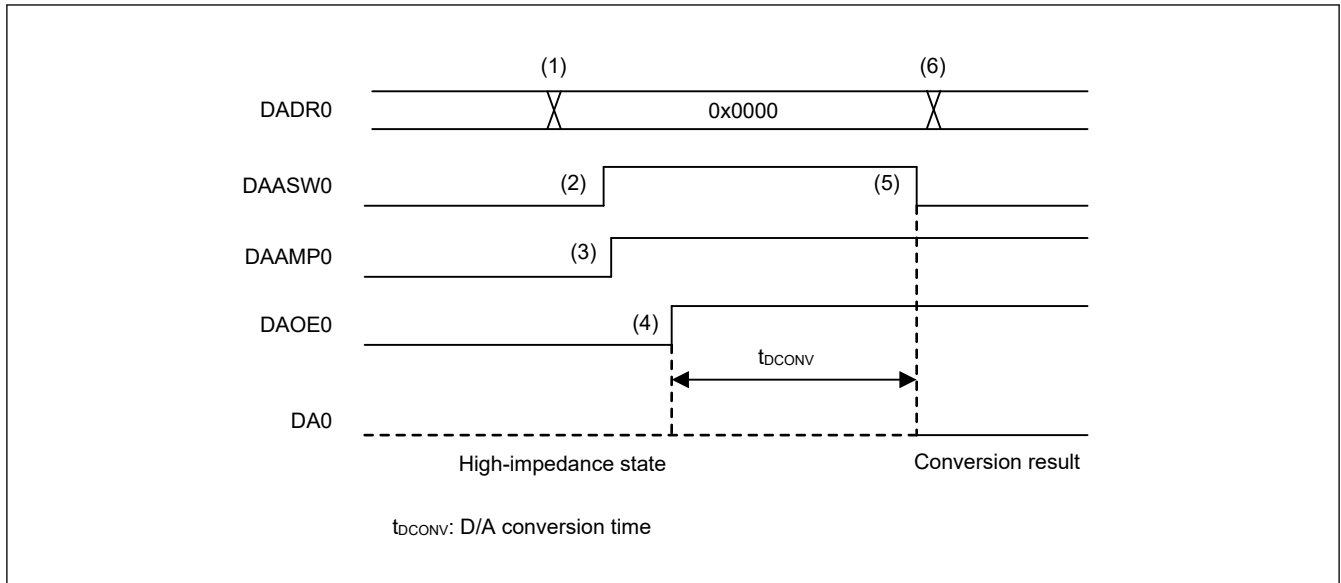
When the MCU enters Deep Software Standby mode with D/A conversion enabled, the outputs of the DAC12 are placed in a high impedance state.

#### 43.7.5 Initialization Procedure with the Output Amplifier

Use the following initialization procedures with the output amplifier. The example shows the case for channel 0.

To initialize the DAC12 with the output amplifier:

1. Write 0x0000 to the DADR0 register.
2. Set the DAASWCR.DAASW0 bit to 1.
3. Set the DAAMPCR.DAAMP0 bit to 1.
4. Set the DACR.DAOE0 bit to 1 to start operation of the amplifier.
5. Clear the DAASWCR.DAASW0 bit to 0 after waiting for the duration of D/A conversion time  $t_{DCONV}$ .
6. Write the value to be converted in the DADR0 register.



**Figure 43.5 Example of the initial flow with the output amplifier in DAC12**

While the amplifier is running, clearing the DACR.DAOE0 bit to 0 allows the amplifier to stop operation. To use the amplifier again, repeat steps 1 to 6.

#### 43.7.6 Initialization Procedure of the Output to internal modules

Use the following initialization procedures for the output to internal modules.

1. Set the DAASWCR.DAASW0 bit to 1.
2. Set the DACR.DAOE0 bit to 1.
3. Write the value to be converted in the DADR0 register.

When you stop output of the DAC12, set DAASWCR.DAASW0 to 0 after setting DACR.DAOE0 to 0.

#### 43.7.7 Constraint on Usage When Interference Reduction between D/A and A/D Conversion Is Enabled

When the DAADSCR.DAADST bit is 1, enabling interference reduction between D/A and A/D conversion, do not place the ADC12 in the module-stop state. Doing so can halt D/A conversion in addition to A/D conversion.



## 44. Temperature Sensor (TSN)

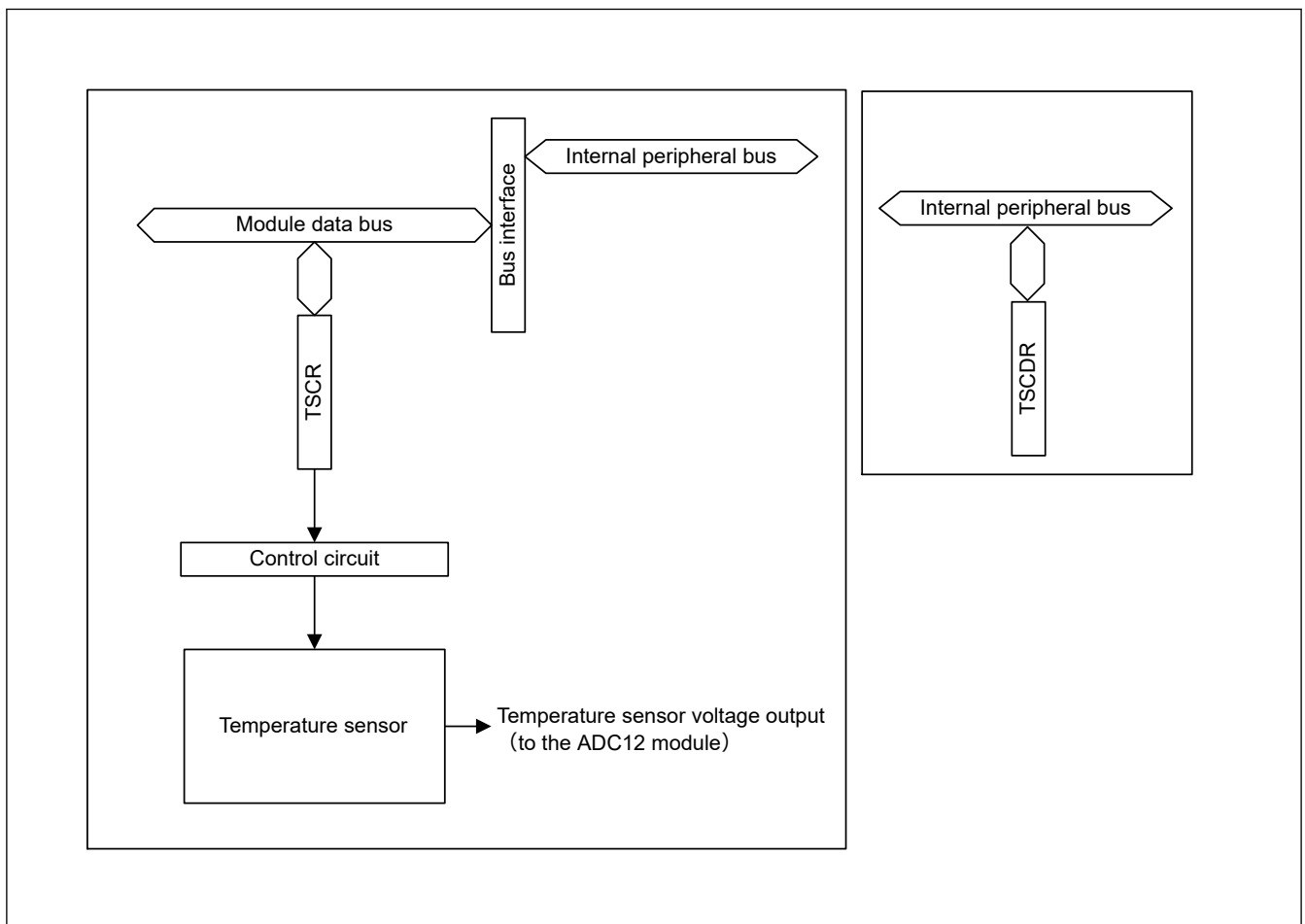
### 44.1 Overview

The on-chip Temperature Sensor (TSN) determines and monitors the die temperature for reliable operation of the device. The sensor outputs a voltage directly proportional to the die temperature, and the relationship between the die temperature and the output voltage is fairly linear. The output voltage is provided to the ADC12 for conversion and can be further used by the end application.

Table 44.1 lists the TSN specifications, and Figure 44.1 shows a block diagram.

**Table 44.1 TSN specifications**

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 12-bit A/D converter
Module-stop function	Module-stop state can be set to reduce power consumption
Temperature sensor calibration data	Reference data measured for each chip at factory shipment is stored in a register
TrustZone Filter	Security and Privilege attribution can be set



**Figure 44.1 TSN block diagram**

## 44.2 Register Descriptions

### 44.2.1 TSCR : Temperature Sensor Control Register

Base address: TSN = 0x4023\_5000  
 TSN\_NS = 0x5023\_5000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TSEN	—	—	TSOE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	TSOE	Temperature Sensor Output Enable 0: Disable output from the temperature sensor to the ADC12 1: Enable output from the temperature sensor to the ADC12	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	TSEN	Temperature Sensor Enable 0: Stop the temperature sensor 1: Start the temperature sensor.	R/W

Note: S-TYPE3, P-TYPE3

The TSCR is a register which controls the temperature sensor. The timing constraints shown in [Figure 44.3](#) apply to the settings of the TSCR register.

#### TSOE bit (Temperature Sensor Output Enable)

The TSOE bit enables or disables the temperature sensor output to ADC12.

#### TSEN bit (Temperature Sensor Enable)

The TSEN bit starts or stops the temperature sensor.

### 44.2.2 TSCDR : Temperature Sensor Calibration Data Register

Base address: TSD = 0x4011\_B000  
 TSD\_NS = 0x5011\_B000

Offset Address: 0x017C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSCDR[15:0]															
Value after reset:	Chip-specific value															

Bit	Symbol	Function	R/W
15:0	TSCDR[15:0]	Temperature Sensor Calibration Data Chip-specific value	R
31:16	—	These bits are read as 0.	R

Note: S-TYPE5, P-TYPE5

The TSCDR register stores temperature sensor calibration data measured for each chip at factory shipment.

Temperature sensor calibration data is the output voltage of the temperature sensor under the conditions  $T_j = 125^\circ\text{C}$  and  $AVCC0 = VREFH0 = 3.3\text{ V}$  converted to a digital value by the 12-bit A/D converter.

The TSCDR register is a read-only 32-bit register. Read from this register in 32-bit units.

Temperature sensor calibration data is stored in the lower 12 bits of the TSCDR register.

### 44.3 Using the Temperature Sensor

The temperature sensor outputs a voltage that varies with the temperature. This voltage is converted to a digital value by the 12-bit A/D converter. To obtain the die temperature, convert this value into the temperature.

#### 44.3.1 Preparation for Using the Temperature Sensor

The ambient temperature (T) is proportional to the temperature sensor voltage output ( $V_s$ ), so ambient temperature is calculated with the following formula:

$$T = (V_s - V_1) / \text{slope} + T_1$$

- T: Ambient temperature of MCU as calculation result ( $^\circ\text{C}$ )
- $V_s$ : Voltage output by the temperature sensor on temperature measurement (V)
- $T_1$ : Temperature experimentally measured at one point ( $^\circ\text{C}$ )
- $V_1$ : Voltage output by the temperature sensor on measurement of  $T_1$  (V)
- $T_2$ : Temperature experimentally measured at a second point ( $^\circ\text{C}$ )
- $V_2$ : Voltage output by the temperature sensor on measurement of  $T_2$  (V)
- Slope: Temperature gradient of the temperature sensor ( $\text{V} / ^\circ\text{C}$ ),  $\text{slope} = (V_2 - V_1) / (T_2 - T_1)$

Characteristics vary between sensors, so Renesas recommends measuring two different sample temperatures as follows:

1. Use the 12-bit A/D converter to measure the voltage  $V_1$  output by the temperature sensor at temperature  $T_1$ .
2. Again use the 12-bit A/D converter to measure the voltage  $V_2$  output by the temperature sensor at a different temperature  $T_2$ .
3. Obtain the temperature gradient ( $\text{slope} = (V_2 - V_1) / (T_2 - T_1)$ ) from these results.
4. Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic ( $T = (V_s - V_1) / \text{slope} + T_1$ ).

In order to make the temperature  $T_a$  and  $T_j$  as close as possible, perform the measurement with the power consumption of the MCU as low as possible.

If you are using the temperature gradient given in [section 52, Electrical Characteristics](#), use the A/D converter to measure the voltage  $V_1$  output by the temperature sensor at temperature  $T_1$ , then calculate the temperature characteristic using the following formula:

$$T = (V_s - V_1) / \text{slope} + T_1$$

Note: This method produces less accurate temperatures than measurement at two points.

In this MCU, the TSCDR register stores the temperature value (CAL125) of the temperature sensor measured under the condition  $T_a = T_j = 125^\circ\text{C}$  and  $AVCC0 = VREFH0 = 3.3\text{ V}$ . If you use this value as the sample measurement result at the first point, you can omit the preparation before using the temperature sensor.

$V_1$  is calculated from CAL125:

$$V_1 = 3.3 \times \text{CAL125} / 4096 \text{ [V]} \text{ (In case of 12 bit accuracy)}$$

Using this value, the measured temperature can be calculated according to the following formula:

$$T = (V_s - V_1) / \text{slope} + 125 \text{ [}^\circ\text{C]}]$$

- T: Ambient temperature of MCU as calculation result ( $^\circ\text{C}$ )
- $V_s$ : Voltage output by the temperature sensor when the temperature is measured (V)
- $V_1$ : Voltage output by the temperature sensor when  $T_a = T_j = 125^\circ\text{C}$  and  $AVCC0 = VREFH0 = 3.3\text{ V}$  (V)

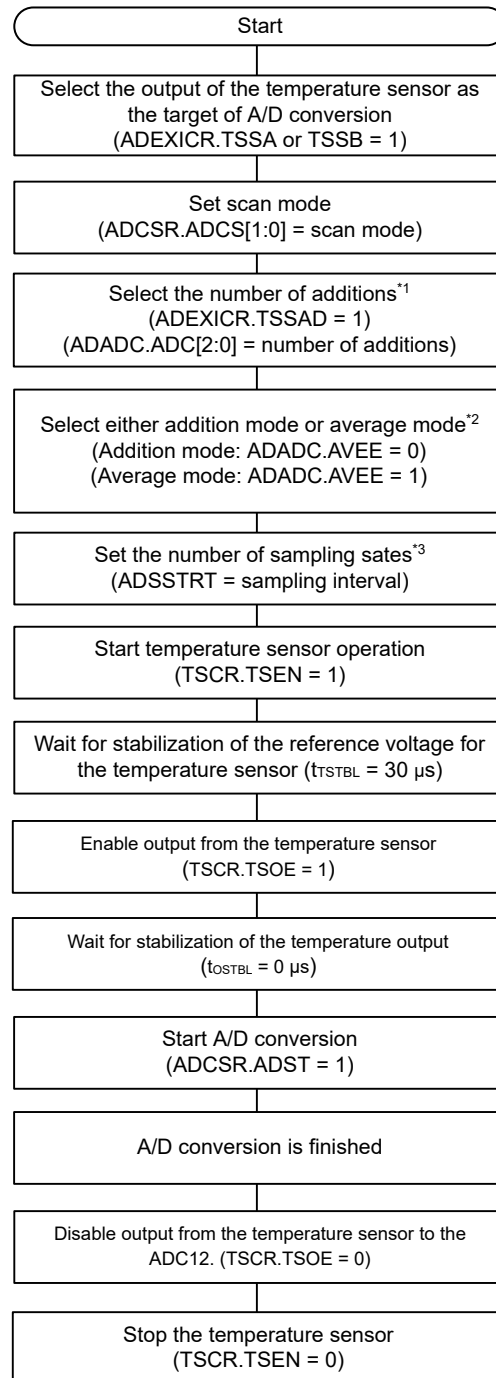
- Slope: Temperature gradient of the temperature sensor<sup>\*1</sup> / 1000 (V/°C)

Note 1. See [section 52, Electrical Characteristics](#)

### 44.3.2 Procedures for Using the Temperature Sensor

[Figure 44.2](#) shows the procedure for using the TSN.

For details, see [section 42, 12-Bit A/D Converter \(ADC12\)](#).



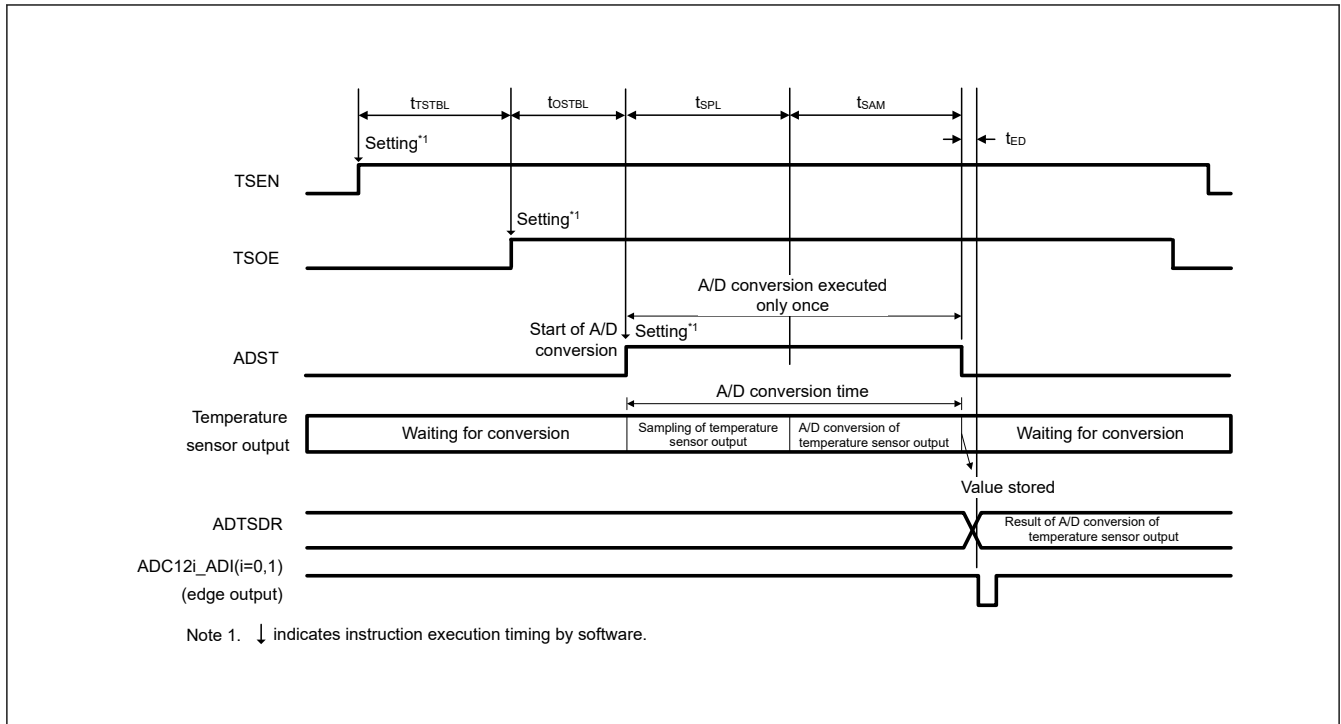
Note 1. This setting is not required if addition/average mode is not set.

Note 2. The ADADC.ADC[2:0] setting is limited to some values in additions/average mode. For details on the available ADADC.ADC[2:0] settings in additions/average mode, see [section 42, 12-Bit A/D Converter \(ADC12\)](#).

Note 3. Set the sampling time to more than the value described in [section 52, Electrical Characteristics](#).

**Figure 44.2 Procedure example for using the TSN**

[Figure 44.3](#) shows the timing from the start of temperature sensor operation until the completion of A/D conversion when the ADC12 is in single scan mode (the conversion target is the temperature sensor output only). The times shown in the figure are described in [Table 44.2](#)



**Figure 44.3** Timing from start of temperature sensor operation until completion of A/D conversion

**Table 44.2** Time until completion of A/D conversion after start of temperature sensor operation

Parameter	Symbol	Time
Wait time for temperature sensor reference voltage stabilization	$t_{STBL}$	30 $\mu$ s (min)
Wait time for temperature sensor output stabilization	$t_{OSTBL}$	0 $\mu$ s (min)
A/D converter input sampling time	$t_{SPL}$	ADSSTRn setting $\times$ ADCLK period
A/D conversion time	$t_{SAM}$	See the table in <a href="#">section 42.3.6. Analog Input Sampling and Scan Conversion Time</a> .
Scan conversion end delay	$t_{ED}$	

## 44.4 Usage Notes

### 44.4.1 Settings for the Module-Stop Function

TSN operation can be disabled or enabled using the associated bit in Module Stop Control Register D (MSTPCRD). The TSN is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

### 44.4.2 Settings for the Software Standby Mode

Before entering Software Standby Mode, set TSCR.TSOE bit to 0 after stopping A/D conversion, then set TSCR.TSEN bit to 0 to stop the TSN.

## 45. High-Speed Analog Comparator (ACMPHS)

### 45.1 Overview

The High-Speed Analog Comparator (ACMPHS) can be used to compare an analog input voltage with a reference voltage and to provide a digital output based on the result of conversion. Both the analog input voltage and the reference voltage can be provided to the ACMPHS from internal sources (D/A converter output or internal reference voltage) and an external source. Such flexibility is useful in applications that require go/no-go comparisons to be performed between analog signals without necessarily requiring A/D conversion.

[Table 45.1](#) lists the ACMPHS specifications, [Figure 45.1](#) shows a block diagram, and [Table 45.2](#) shows the input source configurations.

**Table 45.1 ACMPHS specifications**

Parameter	Specifications
Number of channels	2 channels: ACMPHSn (n = 0, 1)
Analog input voltage	<ul style="list-style-type: none"> <li>Output from internal D/A converter</li> <li>Input from an external source (compatible with internal A/D converter input pin (one selectable))</li> </ul>
Reference voltage	<ul style="list-style-type: none"> <li>Internal reference voltage (Vref)</li> <li>Output from internal D/A converter</li> <li>Input from an external source (compatible with internal A/D converter input pin (one selectable))</li> </ul>
ACMPHS output	<ul style="list-style-type: none"> <li>Comparison result output to terminal PIN</li> <li>Generation of ELC event output</li> <li>Monitor output from register</li> <li>Generation of interrupt request output</li> </ul>
Interrupt request signal	<ul style="list-style-type: none"> <li>Interrupt request generated on valid edge detection from comparison result (when CMPCTL.CSTEN = 0<sup>*1</sup>)</li> <li>Selectable to rising edge, falling edge, or both edges (when CMPCTL.CSTEN = 0<sup>*1</sup>)</li> <li>Rising edge only (when CMPCTL.CSTEN = 1<sup>*1</sup>)</li> </ul>
Digital filter function	<ul style="list-style-type: none"> <li>Selectable to one of three sampling frequencies</li> <li>Not using the filter function is selectable</li> </ul>
Module-stop function	Module-stop state can be set for each groups to reduce power consumption.
TrustZone Filter	Security and Privilege attribution can be set for each group

Note 1. Interrupt request signal selection restriction detail refers to [section 45.5. ACMPHS Interrupts](#) and [section 45.6. ACMPHS Output to the Event Link Controller \(ELC\)](#).

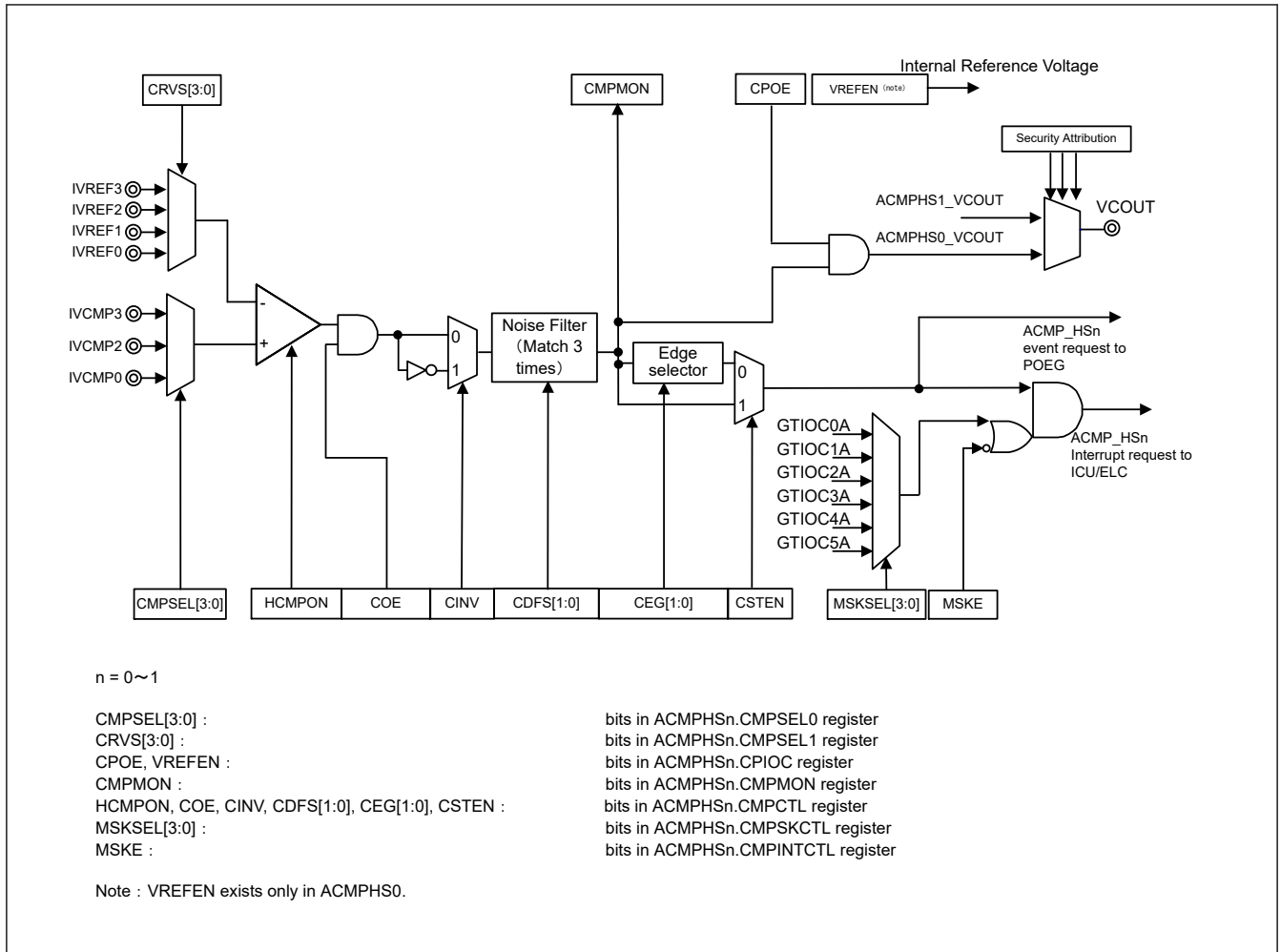


Figure 45.1 ACMPHS block diagram

Table 45.2 Input source configuration of the ACMPHS

Comparator	Reference voltage input source				Analog voltage input source			Output pin
	IVREF3	IVREF2	IVREF1	IVREF0	IVCMP3	IVCMP2	IVCMP0	
ACMPHS0	DA0 <sup>*3</sup>	Vref <sup>*1</sup>	AN104	AN101	AN002	AN000	AN005	VCOUT <sup>*2</sup>
ACMPHS1	DA0 <sup>*3</sup>	Vref <sup>*1</sup>	AN104	AN101	AN102	AN100	—	

Note 1. Internal voltage reference.

Note 2. ACMPHS0 to ACMPHS1 compare outputs are bundled with the VCOUT pin.

Note 3. When D/A converter 0 output is not used, the signal can be used as AN007 analog input.

## 45.2 Register Descriptions

### 45.2.1 CMPCTL : Comparator Control Register

Base address: ACMPHSn = 0x4023\_6000 + 0x0100 × n (n = 0, 1)  
 ACMPHSn\_NS = 0x5023\_6000 + 0x0100 × n (n = 0, 1)

Offset address: 0x000

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HCMPON	CDFS[1:0]		CEG[1:0]		CSTEN	COE	CINV

Value after reset: 0 0 0 0 0 0 0 0 0



Bit	Symbol	Function	R/W
0	CINV	Comparator Output Polarity Selection*1 *2 0: Do not invert comparator output 1: Invert comparator output	R/W
1	COE	Comparator Output Enable 0: Disable comparator output (output signal is low level) 1: Enable comparator output	R/W
2	CSTEN	Interrupt Select*3 0: Output through the edge selector 1: Output directly	R/W
4:3	CEG[1:0]	Selection of Valid Edge (Edge Selector) 0 0: Do not detect edge 0 1: Detect rising edge 1 0: Detect falling edge 1 1: Detect both edges	R/W
6:5	CDFS[1:0]	Noise Filter Selection *1 *2 *3 *4 0 0: Do not use noise filter 0 1: Use noise filter sampling frequency of PCLKB/2 <sup>3</sup> 1 0: Use noise filter sampling frequency of PCLKB/2 <sup>4</sup> 1 1: Use noise filter sampling frequency of PCLKB/2 <sup>5</sup>	R/W
7	HCOMPON	Comparator Operation Control*5 0: Stop operation (comparator outputs a low-level signal) 1: Enable operation (enables input to the comparator pins)	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Disable the ACMPHS output (COE = 0) before changing the CDFS[1:0] and CINV bits.

Note 2. If the CDFS[1:0] and CINV bits are changed, an ACMPHS interrupt request and an ELC event might be generated. Before changing these bits, set the ELSRn register to 0 (the ACMPHS output is not linked). After changing these bits, clear the IR flag in the IELSRn register to 0 to clear the interrupt status.

Note 3. Set the CSTEN bit to 1 and the CDFS[1:0] bits to 00b if the ACMPHS interrupt causes release of Software Standby mode. CSTEN is supported only by the ACMPHS0. ACMPHS1.CMPCTL.CSTEN must be set to 0.

Note 4. If the CDFS[1:0] bits are changed from 00b (noise filter not used) to a value other than 00b (noise filter used), perform sampling four times and update the filter output, and then use the ACMPHS interrupt request or the ELC event.

Note 5. A stabilization wait time is required to permit ACMPHS operation after enabling it (HCOMPON = 1). The operation stabilization wait time for ACMPHS modules 0 to 1 is 300 ns.

Note: Set this register before setting registers in the POEG when using comparator output as a POEG source.

The CMPCTL register controls the ACMPHS operation, enables or disables the ACMPHS output, selects the noise filter, selects the valid edge of the interrupt signal, and selects the interrupt.

## 45.2.2 CMPSEL0 : Comparator Input Select Register

Base address: ACMPHSn = 0x4023\_6000 + 0x0100 × n (n = 0, 1)  
ACMPHSn\_NS = 0x5023\_6000 + 0x0100 × n (n = 0, 1)

Offset address: 0x004

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	CMPSEL[3:0]		
------------	---	---	---	---	-------------	--	--

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
3:0	CMPSEL[3:0]	Comparator Input Selection*1 *2 *3 0x00: Do not input 0x01: Select IVCMP0 0x04: Select IVCMP2 0x08: Select IVCMP3 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Use the following procedure to change the CMPSEL[3:0] bits. Writing a value other than 0x00 while the value of the CMPSEL0 register is not 0x00 is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.

To change the CMPSEL[3:0] bits:

1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL0 register to 0x00.
3. Set a new value in the CMPSEL[3:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time (200 ns).
5. Set the CMPCTL.COE bit to 1.
6. Clear the IR flag in the IELSRn register to clear the interrupt status.

Note 2. For details, see [Table 45.2](#).

Note 3. If ACMPHSn level detection signal is used as a POEG source, write access to these bits after the setting of any register in the POEG may generate a POEG source.

### 45.2.3 CMPSEL1 : Comparator Reference Voltage Select Register

Base address: ACMPHSn = 0x4023\_6000 + 0x0100 × n (n = 0, 1)  
ACMPHSn\_NS = 0x5023\_6000 + 0x0100 × n (n = 0, 1)

Offset address: 0x008

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CRVS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CRVS[3:0]	Reference Voltage Selection*1*2*3*4 0x00: Do not input 0x01: Select IVREF0 0x02: Select IVREF1 0x04: Select IVREF2 0x08: Select IVREF3 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. Use the following procedure to change the CRVS[3:0] bits. Writing a value other than 0x00 while the value of the CMPSEL1 register is not 0x00 is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.

To change the CRVS[3:0] bits:

1. Set the CMPCTL.COE bit to 0.
2. Set the CMPSEL1 register to 0x00.
3. Set a new value to the CRVS[3:0] bits, with 1 set in only one of the bits.
4. Wait for the input switching stabilization wait time (200 ns).
5. Set the CMPCTL.COE bit to 1.
6. Clear the IR flag in the IELSRn register to clear the interrupt status.

Note 2. For details, see [Table 45.2](#).

Note 3. When the on-chip D/A converter output voltage is used, set the D/A converter to generate comparator C reference voltage before enabling comparator operation (CMPCTL.HCMPON bit = 1). For details on setting the D/A converter, see [section 43, 12-Bit D/A Converter \(DAC12\)](#).

Note 4. If ACMPHSn level detection signal is used as a POEG source, write access to these bits after the setting of any register in the POEG may generate a POEG source.

### 45.2.4 CMPMON : Comparator Output Monitor Register

Base address: ACMPHSn = 0x4023\_6000 + 0x0100 × n (n = 0, 1)  
ACMPHSn\_NS = 0x5023\_6000 + 0x0100 × n (n = 0, 1)

Offset address: 0x00C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	COMP MON
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	COMPON	Comparator Output Monitor* <sup>1</sup> 0: Comparator output is low 1: Comparator output is high	R
7:1	—	These bits are read as 0. The write value should be 0.	R

Note: S-TYPE-3, P-TYPE-3

Note 1. When ACMPHS operation is enabled (CMPCTL.HCMPON = 1 and CMPCTL.COE = 1) but the noise filter is not in use (CDFS[1:0] = 00b), design the software so that the COMPON bit is read twice and the values are only used after the two consecutive values match.

#### 45.2.5 CPIOC : Comparator Output Control Register

Base address:  $ACMPHSn = 0x4023\_6000 + 0x0100 \times n$  (n = 0, 1)  
 $ACMPHSn\_NS = 0x5023\_6000 + 0x0100 \times n$  (n = 0, 1)

Offset address: 0x010

Bit position: 7 6 5 4 3 2 1 0

Bit field:	VREF EN	—	—	—	—	—	—	CPOE
------------	------------	---	---	---	---	---	---	------

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	CPOE	External Pin Output Enable Comparison result by the comparator is output to an external pin. 0: Output to the comparator external pin is disabled (the output signal is fixed to low) 1: Output to the comparator external pin is enabled	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	VREFEN	Internal Vref Enable* <sup>1</sup> 0: Disable internal Vref 1: Enable internal Vref	R/W

Note: S-TYPE-3, P-TYPE-3

Note 1. For ACMPHS modules 0 to 1, VREFEN exists only in ACMPHS0.CPIOC. When using the internal Vref in ACMPHS0 to ACMPHS1, set the VREFEN bit in ACMPHS0.CPIOC to 1. Bit [7] in ACMPHS1.CPIOC should be 0 regardless of whether or not the internal Vref is used.

#### 45.2.6 CPINTCTL : Comparator Interrupt Control Register

Base address:  $ACMPHSn = 0x4023\_6000 + 0x0100 \times n$  (n = 0, 1)  
 $ACMPHSn\_NS = 0x5023\_6000 + 0x0100 \times n$  (n = 0, 1)

Offset address: 0x040

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	—	—	MSKE
------------	---	---	---	---	---	---	---	------

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	MSKE	Comparator Interrupt Periodic Mask Enable 0: Disable interrupt masking (Default) 1: Enable interrupt masking by GPT output signal selected by CPMSKCTL.MSKSEL[2:0]	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register controls the interrupt to ICU/ELC, while the event to POEG(Port Output Enable) is not controlled by this register.

## 45.2.7 CPMSKCTL : Comparator Interrupt Mask Control Register

Base address:  $ACMPHSn = 0x4023\_6000 + 0x0100 \times n$  ( $n = 0, 1$ )  
 $ACMPHSn\_NS = 0x5023\_6000 + 0x0100 \times n$  ( $n = 0, 1$ )

Offset address: 0x044

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	MSKSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	MSKSEL[2:0]	Comparator Interrupt Periodic Mask Selection 0 0 0: Enable interrupt masking by GTIOC0A output signal 0 0 1: Enable interrupt masking by GTIOC1A output signal 0 1 0: Enable interrupt masking by GTIOC2A output signal 0 1 1: Enable interrupt masking by GTIOC3A output signal 1 0 0: Enable interrupt masking by GTIOC4A output signal 1 0 1: Enable interrupt masking by GTIOC5A output signal Others: Setting prohibited.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

This register controls the interrupt to ICU/ELC, while the event to POEG(Port Output Enable) is not controlled by this register.

## 45.3 Operation

The ACMPHS compares a reference voltage to an analog input voltage. Operation is not guaranteed when the values of registers are changed during ACMPHS operation. Table 45.3 shows the procedures for setting the registers associated with ACMPHS.

**Table 45.3 Procedure for setting registers associated with ACMPHSn (n = 0, 1) (1 of 2)**

Step	Register	Bit	Setting
1	Associated MSTPCRD register	MSTPD28 to MSTPD27	0: Input clock supply.
2	Associated pin function control register (PFS)	ASEL	1: Select the function of pins IVREF and IVCMP.
3	ACMPHSn.CPIOC	VREFEN	1: When using the internal Vref.
4	Associated D/A convertor		When using the D/A convertor, select in the register.
5	CMPSEL0, CMPSEL1	CMPSEL[3:0] CRVS[3:0]	Select the ACMPHSn input, with 1 set in only one of the bits.
6	CMPCTL	CDFS[1:0], CEG[1:0], and CINV	Set up ACMPHSn control.
		HCOMPON	1: Enable ACMPHSn operation.
7	CPMSKCTL	MSKCTL[2:0]	Select the interrupt mask source signal (from GPT)
8	CPINT	MSKE	1: Enable interrupt mask function if need
9	Waiting for the ACMPHS stabilization time (minimum 300 ns).		
10	CMPCTL	COE	1: Enable ACMPHSn output.
11	CPIOC	CPOE	Set the VCOUT output
	Associated pin function control register (PFS)	PSEL, PMR	Select the VCOUT port function.
12	IELSRn	IR, IELS[8:0]	When using an interrupt, select the interrupt status flag and the ICU event link.*1
13	ELSRn	ELS[8:0]	When using an ELC, select the event link*2.
14	Operation started		

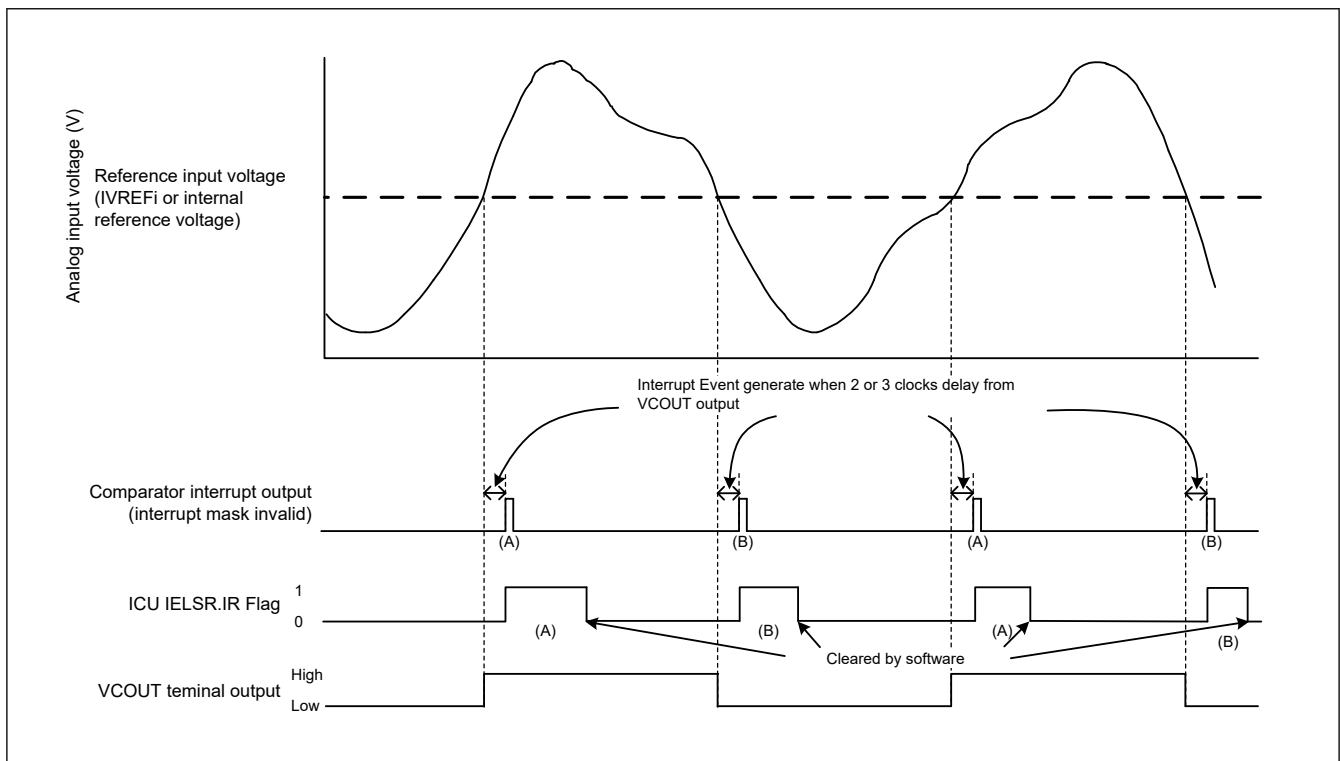
**Table 45.3 Procedure for setting registers associated with ACMPHSn (n = 0, 1) (2 of 2)**

Step	Register	Bit	Setting
15	CMPCTL	COE	0: When changing IVREF or IVCMP, to disable ACMPHSn output.
16	CMPSEL1	CRVS[3:0]	Change the CMPSEL1 bits as follows: 1. Set bits CMPSEL1 to 0000 0000b. 2. Set a new value to the CMPSEL1 bits, with 1 set in only one of the bits.
	CMPSEL0	CMPSEL[3:0]	Change the CMPSEL0 bits as follows: 1. Set bits CMPSEL0 to 0000 0000b. 2. Set a new value to the CMPSEL0 bits, with 1 set in only one of the bits.
17	Waiting for the ACMPHS switching stabilization time (minimum 200 ns).		
18	CMPCTL	COE	1: Enable ACMPHSn output.
19	Operation restarted		

Note 1. After ACMPHSn is set, an unnecessary interrupt might occur until operation becomes stable, so initialize the interrupt flag.

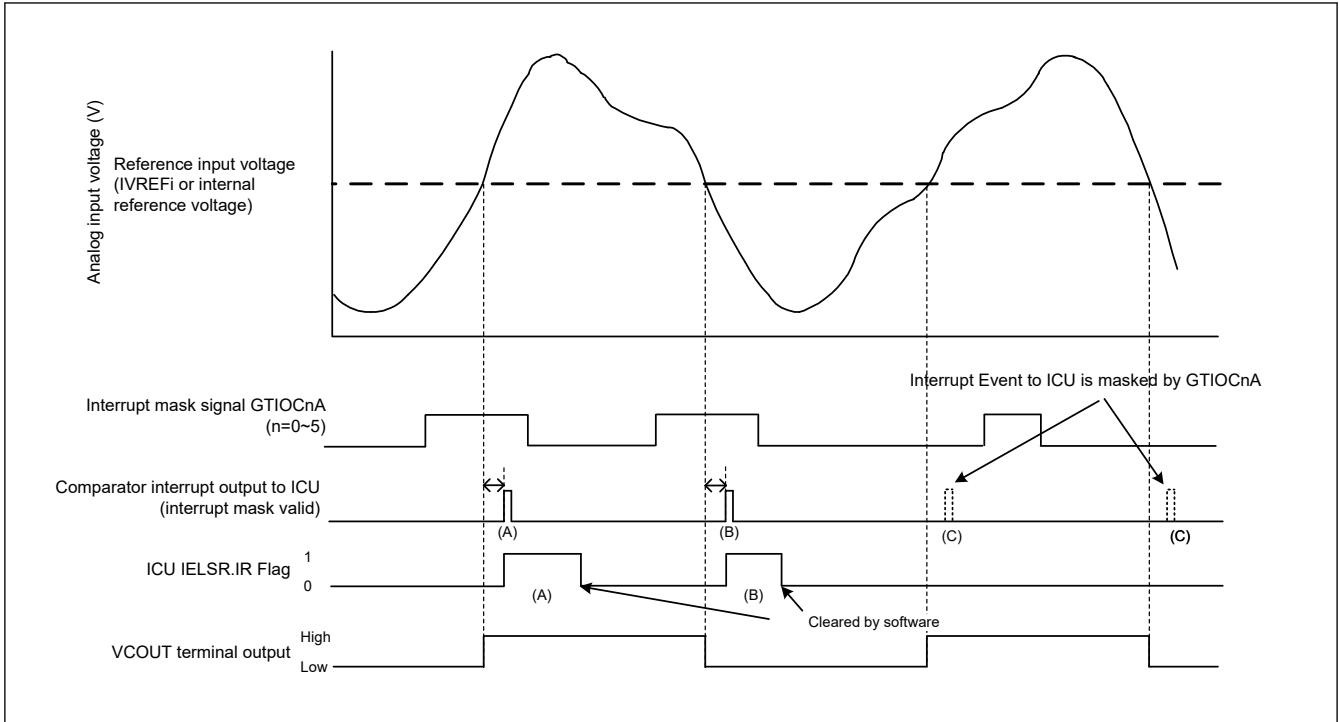
Note 2. After ACMPHSn is set, an unnecessary interrupt might occur until operation becomes stable, so initialize the event link select.

Figure 45.2 and Figure 45.3 show an example of ACMPHS operation. The VCOUT output becomes 1 when the analog input voltage is higher than the ACMPHS reference input voltage, and the VCOUT output becomes 0 when the analog input voltage is lower than the reference voltage. When the ACMPHS output changes, an interrupt request and an ELC event are output.



**Figure 45.2 ACMPHS operation example (interrupt mask function invalid)**

Figure 45.2 applies when CPOE = 1 (pin output enabled), CDFS[1:0] = 00b (filter not used), and CEG[1:0] = 11b (both-edge detection selected). When CINV = 0, CEG[1:0] = 01b (rising-edge detection selected for non-inversion output signal from the ACMPHS), the IELSR.IR flag changes as shown by (A) only. When CINV = 0, CEG[1:0] = 10b (falling-edge detection selected for non-inversion output signal from the ACMPHS), the IR flag changes as shown by (B) only. When CPOE = 1, VCOUT directly outputs.



**Figure 45.3 ACMPHS operation example (interrupt mask function valid)**

Figure 45.3 applies when CPOE = 1 (pin output enabled), CDFS[1:0] = 00b (filter not used), CEG[1:0] = 11b (both-edge detection selected). When CINV = 0, CEG[1:0] = 01b (rising-edge detection selected for non-inversion output signal from the ACMPHS), the IELSR.IR flag changes as shown by (A) only. When CINV = 0, CEG[1:0] = 10b (falling-edge detection selected for non-inversion output signal from the ACMPHS), the IR flag changes as shown by (B) only.

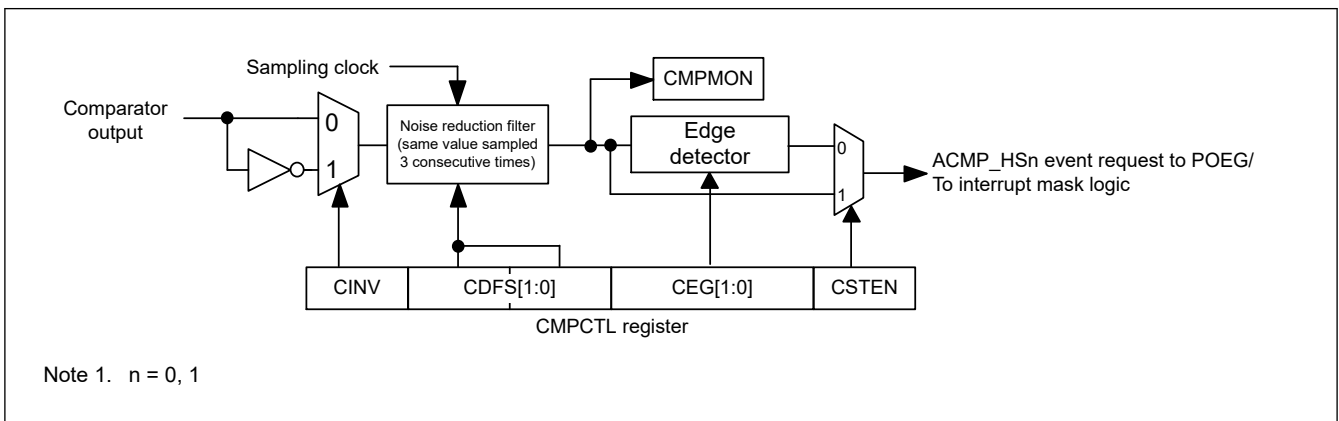
When MSKE = 1 and MSKSEL[2:0] = 000b, the interrupt signal output to ICU is masked by GTIOC0A when GTIOC0A = low level as shown by (C), while it passes through when GTIOC0A = high level.

When CPOE = 1, VCOUT directly outputs, regardless of the MSKE setting.

### 45.4 Noise Filter

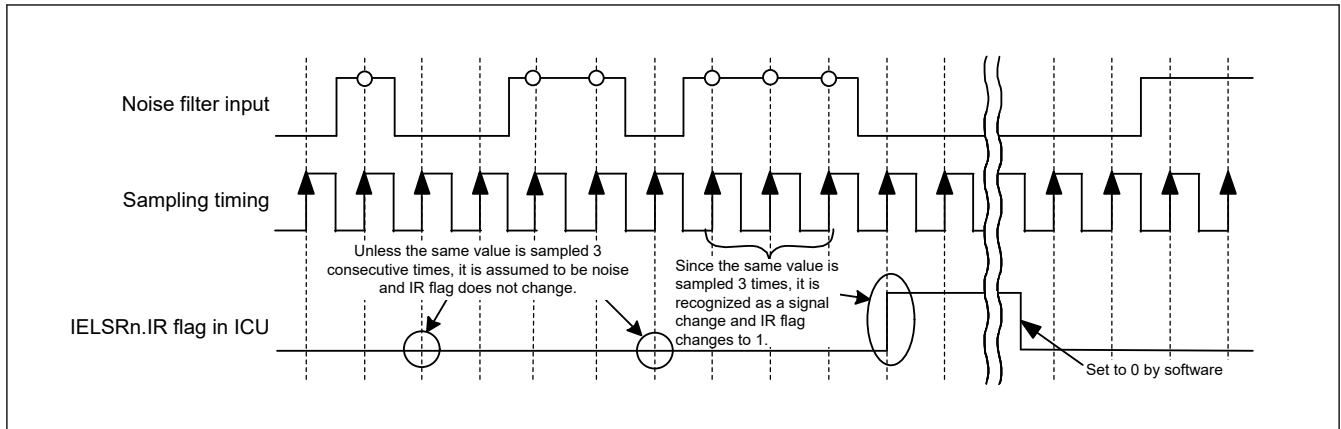
The ACMPHS contains a noise filter. The sampling clock can be selected in the CMPCTL.CDFS[1:0] bits. The comparator output signal is sampled every sampling clock, and if the same value is sampled three times, the noise filter output at the next sampling clock cycle is used as the ACMPHS output.

Figure 45.4 shows the configuration of the noise filter and edge detector, and Figure 45.5 shows an example of noise filter and interrupt operation.



Note 1. n = 0, 1

**Figure 45.4 Noise filter and edge detection configuration**



**Figure 45.5 Noise filter and interrupt operation example**

The operation example in [Figure 45.5](#) applies when the `CMPCTL.CDFS[1:0]` bits are 01b, 10b, or 11b (noise filter used).

## 45.5 ACMPHS Interrupts

The ACMPHS generates two interrupt requests from sources `ACMPHSn` ( $n = 0, 1$ ). To use an ACMPHS interrupt, select it in the `IELSR` register in the Interrupt Controller Unit (ICU). Select the interrupt request in the `CMPCTL.CSTEN` bit either through the edge selector, or not.

Interrupt event to ICU/ELC can be masked by GPT output (`GTIOCnA` ( $n = 0$  to 5)), it's controlled by `CPINTCTL.MSKE` bit and `CPMSKCTL.MSKSEL[2:0]` bits. Detail refer to [Figure 45.3](#).

Event to PORG can't be masked by GPT output (`GTIOCnA` ( $n = 0$  to 5)), it's not controlled by `CPINTCTL.MSKE` bit and `CPMSKCTL.MSKSEL[2:0]` bits.

When using the ACMPHS interrupt through the edge selector, set at least one of the `CMPCTL.CEG[1:0]` bits to 1 (to a value other than 00b for no edge selection). Set the `CMPCTL.CSTEN` bit to 0 (output through the edge selector) in Normal mode, CPU Sleep mode, and CPU Deep Sleep mode.

To use the ACMPHS interrupt in Software Standby mode, set the `CMPCTL.CSTEN` bit to 1 (direct output), set the `CMPCTL.CDFS[1:0]` bit to 00b (digital noise filter not used), and set `CMPCTL.CINV` as follows:

- When detecting compare result 0 to 1, set `CMPCTL.CINV` to 0 (comparator output not inverted)
- When detecting compare result 1 to 0, set `CMPCTL.CINV` to 1 (comparator output inverted).

An `ACMPHS0` interrupt request can be used to release Software Standby mode. `ACMPHS1` cannot be used.

`ACMPHS0/ACMPHS1` both can not be used in Deep Software Standby mode.

For details on the register settings related to ACMPHS interrupt requests, see [section 45.2.1. `CMPCTL` : Comparator Control Register](#), [section 45.2.6. `CPINTCTL` : Comparator Interrupt Control Register](#) and [section 45.2.7. `CPMSKCTL` : Comparator Interrupt Mask Control Register](#).

## 45.6 ACMPHS Output to the Event Link Controller (ELC)

The ELC uses the ACMPHS interrupt request signal as an ELC event signal, enabling link operation for the preset module. To use the ACMPHS ELC event, select them in the `ELSR` register in the ELC. When using the ELC event request, set the `CMPCTL.CSTEN` bit to 0 (output through the edge selector). Also set at least one of the `CMPCTL.CEG[1:0]` bits to 1 (to a value other than 00b for no edge selection).

Interrupt event to ELC can be masked by GPT output (`GTIOCnA` ( $n = 0$  to 5)), it's controlled by `CPINTCTL.MSKE` bit and `CPMSKCTL.MSKSEL[2:0]` bits.

## 45.7 ACMPHS Pin Output

The comparison result from the ACMPHS can be output to external pins. Use the `CMPCTL.CINV` and `CPIOC.CPOE` bits to set the output polarity (non-inverted or inverted output) and enable or disable output. To output the ACMPHS comparison result to the `VCOUT` output pin, set the associated port `mn` pin function control register (`PmnPFS`) in the I/O register.

Only when the security attribution of Port and ACMPHS match, the ACMPHS comparison result is output to the VCOUT pin as shown in the [Table 45.4](#).

**Table 45.4 The output condition of VCOUT pin by the security setting**

PORT Security Attribution (PmSAR (m = 0 to 9))	ACMPHS0 Security Attribution PSARD	ACMPHS1 Security Attribution PSARD	VCOUT pin output
0 (secure)	0 (secure)	0 (secure)	both ACMPHS0/1 output enable
1 (non-secure)	0 (secure)	0 (secure)	both ACMPHS0/1 output disable
0 (secure)	0 (secure)	1 (non-secure)	ACMPHS0_VCOOUT output enable ACMPHS1_VCOOUT output disable
1 (non-secure)	0 (secure)	1 (non-secure)	ACMPHS0_VCOOUT output disable ACMPHS1_VCOOUT output enable
0 (secure)	1 (non-secure)	0 (secure)	ACMPHS0_VCOOUT output disable ACMPHS1_VCOOUT output enable
1 (non-secure)	1 (non-secure)	0 (secure)	ACMPHS0_VCOOUT output enable ACMPHS1_VCOOUT output disable
0 (secure)	1 (non-secure)	1 (non-secure)	both ACMPHS0/1 output disable
1 (non-secure)	1 (non-secure)	1 (non-secure)	both ACMPHS0/1 output enable

## 45.8 Usage Notes

### 45.8.1 Settings for the Module-Stop Function

ACMPHS operation can be disabled or enabled using the Module Stop Control Register. The ACMPHS is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

### 45.8.2 Settings for the DAC12

ACMPHS is connected to the internal module output of the D/A converter. For details, see [section 43, 12-Bit D/A Converter \(DAC12\)](#).

### 45.8.3 Relationship with the ADC12

Constraints apply on the simultaneous use of ACMPHS analog input and ADC12 analog input. For details, see [section 42.6.14, Relationship between ADC12 Units 0 and 1 and the ACMPHS](#).

### 45.8.4 ACMPHS Operation in Module-Stop State

When the module-stop state is entered while ACMPHS is operating, analog circuits in the ACMPHS are not stopped and the analog power supply current is the same as that when ACMPHS is being used. If the analog power supply current needs to be reduced in the module-stop state, set the CMPCTL.HCMPON bit to 0 to stop the ACMPHS.

### 45.8.5 ACMPHS Operation in Software Standby Mode

When Software Standby mode is entered while ACMPHS is operating, analog circuits in the ACMPHS are not stopped and the analog power supply current is the same as that when ACMPHS is being used. If the analog power supply current needs to be reduced in Software Standby mode, set the CMPCTL.HCMPON bit to 0 to stop the ACMPHS.

### 45.8.6 Setting the D/A Converter for Generating Reference Voltage

Set the D/A converter to generate reference voltage and wait for the D/A converter output settling time before enabling the comparator. Similarly, before making any changes to the settings of the D/A converter, stop the comparator temporarily. After the changes are made, wait for the D/A converter output settling time before enabling the comparator.



## 46. Data Operation Circuit (DOC)

This is the DOC\_B version of the DOC peripheral module.

DOC\_B is referred to as DOC in this chapter.

### 46.1 Overview

The data operation circuit (DOC) is used to compare, add, and subtract 16 or 32-bit data. An interrupt can be generated when the following conditions apply.

- When the 16 or 32-bit compared values match the detection condition
- When the result of 16 or 32-bit data addition overflows
- When the result of 16 or 32-bit data subtraction underflows

Table 46.1 lists the data operation circuit specifications and Figure 46.1 shows a block diagram of the data operation circuit.

**Table 46.1 DOC specifications**

Item	Description
Data operation function	<ul style="list-style-type: none"> <li>• 16 or 32-bit data comparison, comparison to detect data above or below thresholds, and window comparison</li> <li>• 16 or 32-bit data addition, and subtraction</li> </ul>
Module-stop function	The module-stop state can be set to reduce power consumption.
Interrupts	<ul style="list-style-type: none"> <li>• The compared values match the detection condition</li> <li>• The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1)</li> <li>• The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)</li> </ul>
Event link function (output)	<ul style="list-style-type: none"> <li>• The result of data comparison is consistent with detection condition</li> <li>• The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1)</li> <li>• The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1)</li> </ul>
TrustZone Filter	Security and Privilege attribution can be set

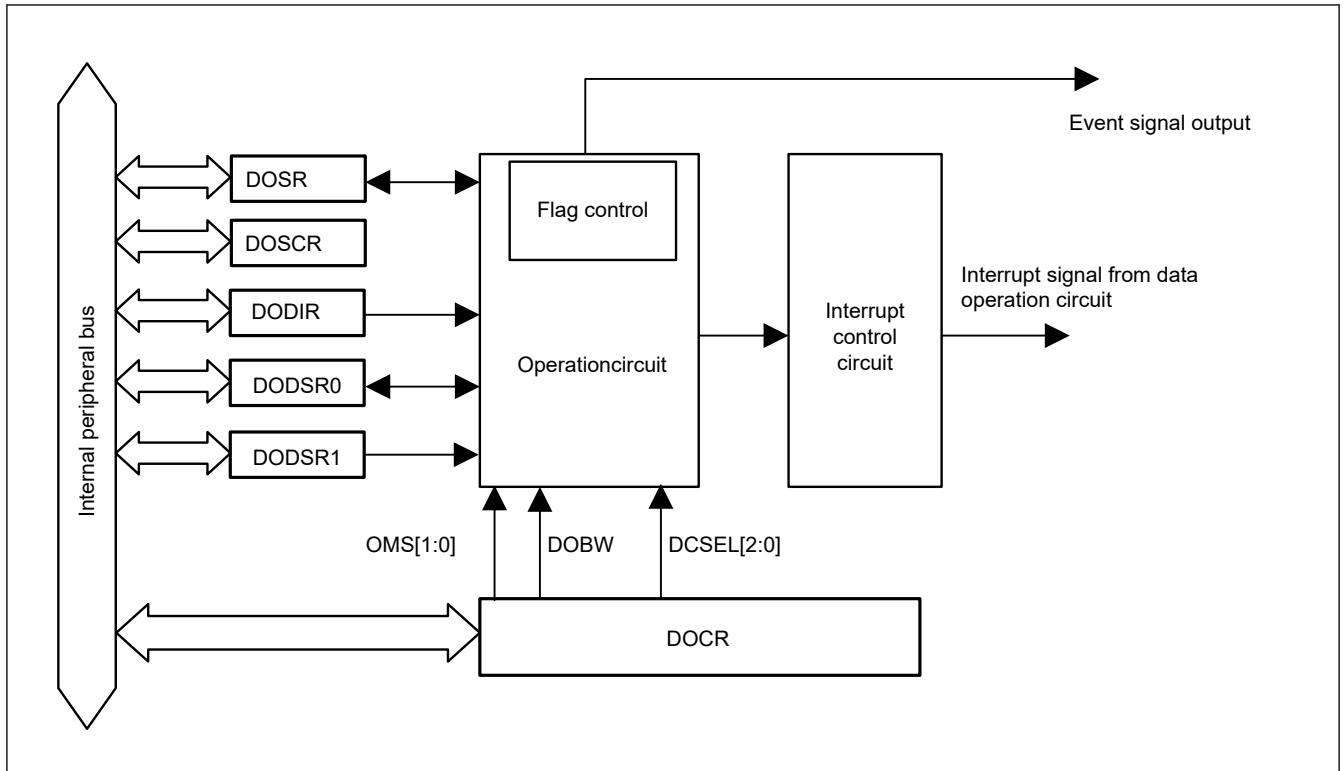


Figure 46.1 DOC block diagram

## 46.2 DOC Register Descriptions

### 46.2.1 DOCR : DOC Control Register

Base address: DOC\_B = 0x4031\_1000  
 DOC\_B\_NS = 0x5031\_1000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	DCSEL[2:0]		DOBW	—	OMS[1:0]		

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
1:0	OMS[1:0]	Operating Mode Select 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	DOBW	Data Operation Bit Width Select 0: 16-bit 1: 32-bit	R/W
6:4	DCSEL[2:0] <sup>1</sup>	Detection Condition Select 0 0 0: Mismatch (DODSR0 ≠ DODIR) 0 0 1: Match (DODSR0 = DODIR) 0 1 0: Lower (DODSR0 > DODIR) 0 1 1: Upper (DODSR0 < DODIR) 1 0 0: Inside window (DODSR0 < DODIR < DODSR1) 1 0 1: Outside window (DODIR < DODSR0, DODSR1 < DODIR) Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE3, P-TYPE3

Note 1. Valid only when data comparison mode is selected.

The DOCR is a register which can set the operation mode of data operation circuit and interrupt enable/disable.

### OMS[1:0] bits (Operating Mode Select)

These bits select the operating mode of the data operation circuit.

### DOBW bit (Data Operation Bit Width Select)

This bit selects the bit width of data operation.

### DCSEL[2:0] bits (Detection Condition Select)

These bits are valid only when data comparison mode is selected.

These bits select the condition for detection in data comparison mode.

## 46.2.2 DOSR : DOC Flag Status Register

Base address: DOC\_B = 0x4031\_1000  
DOC\_B\_NS = 0x5031\_1000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DOPCF

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	DOPCF	Data Operation Circuit Flag Indicates the result of an operation.	R
7:1	—	These bits are read as 0.	R

Note: S-TYPE3, P-TYPE3

The DOSR register indicates the status of the data operation.

### DOPCF flag (Data Operation Circuit Flag)

[Setting conditions]

- DOCR.OMS[1:0] bits = 00b (Data comparison mode): The compared value matches the detection condition selected by DOCR.DCSEL[2:0] bits
- DOCR.OMS[1:0] bits = 01b (Data addition mode): The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF\_FFFF (DOCR.DOBW = 1)
- DOCR.OMS[1:0] bits = 10b (Data subtraction mode): The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000\_0000 (DOCR.DOBW = 1)

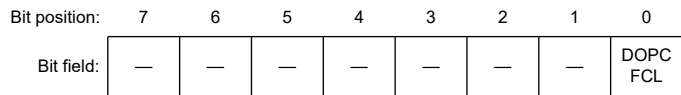
[Clearing condition]

- Writing 1 to the DOSCR.DOPCFCL bit

### 46.2.3 DOSCR : DOC Flag Status Clear Register

Base address: DOC\_B = 0x4031\_1000  
DOC\_B\_NS = 0x5031\_1000

Offset address: 0x08



Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	DOPCFCL	DOPCF Clear 0: Maintains the DOPCF flag state. 1: Clears the DOPCF flag.	W
7:1	—	The write value should be 0.	W

Note: S-TYPE3, P-TYPE3

The DOSCR is a register which can clear the status of data operation. This register is read as 0x00.

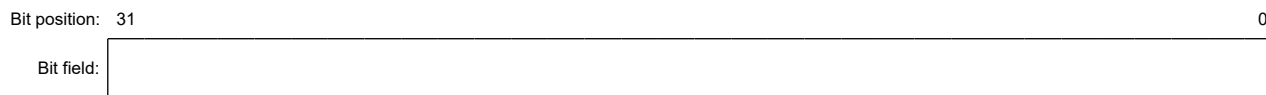
#### DOPCFCL bit (DOPCF Clear)

Setting this bit to 1 clears the DOPCF flag.

### 46.2.4 DODIR : DOC Data Input Register

Base address: DOC\_B = 0x4031\_1000  
DOC\_B\_NS = 0x5031\_1000

Offset address: 0x0C



Value after reset: 0

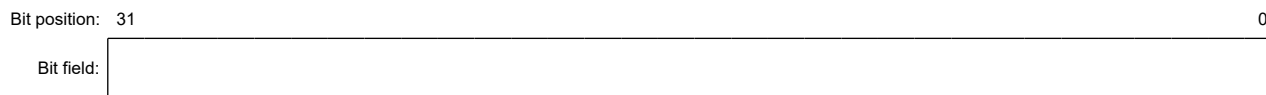
Bit	Symbol	Function	R/W
31:0	n/a	It stores data used in the operations. Access the DODIR with the bit width of data operation selected by the DOCR.DOBW bit.	R/W

Note: S-TYPE3, P-TYPE3

### 46.2.5 DODSR0 : DOC Data Setting Register 0

Base address: DOC\_B = 0x4031\_1000  
DOC\_B\_NS = 0x5031\_1000

Offset address: 0x10



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	Access the DODSR0 with the bit width of data operation selected by the DOCR.DOBW bit. This register stores data for use as a reference in data comparison mode. When selecting window comparison (DOCR.DCSEL[2:0] = 100b, 101b), set a value less than DODSR1 (DODSR1 > DODSR0). This register also stores the results of operations in data addition and data subtraction modes.	R/W

Note: S-TYPE3, P-TYPE3

### 46.2.6 DODSR1 : DOC Data Setting Register 1

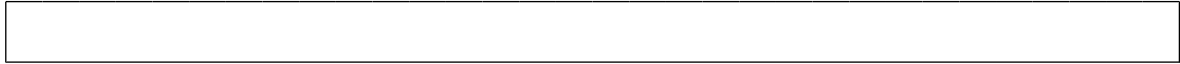
Base address: DOC\_B = 0x4031\_1000  
 DOC\_B\_NS = 0x5031\_1000

Offset address: 0x14

Bit position: 31

0

Bit field:



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	Access the DODSR1 with the bit width of data operation selected by the DOCR.DOBW bit. This register stores data for use as a reference in data comparison mode. When selecting window comparison (DOCR.DCSEL[2:0] = 100b, 101b), set a value greater than DODSR0 (DODSR1 > DODSR0). This register is only used for window comparisons.	R/W

Note: S-TYPE3, P-TYPE3

## 46.3 Operation

### 46.3.1 Data Comparison Mode

Figure 46.2 to Figure 46.7 shows an example of the steps involved in data comparison mode operation by the data operation circuit.

The following is an example of operation when the bit width of data operation is 32-bit.

1. Writing 00b to the DOCR.OMS[1:0] bits selects data comparison mode, and setting the DOCR.DCSEL[2:0] to selects detection condition.
2. The 32-bit reference data is set in DODSR0 and DODSR1.\*1
3. 32-bit data for comparison is written to DODIR.
4. If a value written to DODIR match the detection condition set by DOCR.DCSEL[2:0], the DOCR.DOPCF flag is set to 1 and an ELC event and a data operation circuit interrupt are generated.

Note: The comparison operation is executed only by writing to the DODIR

Note 1. The DODSR1 register setting is required only when window comparison is selected. Set a value greater than DODSR0 (DODSR1 > DODSR0).

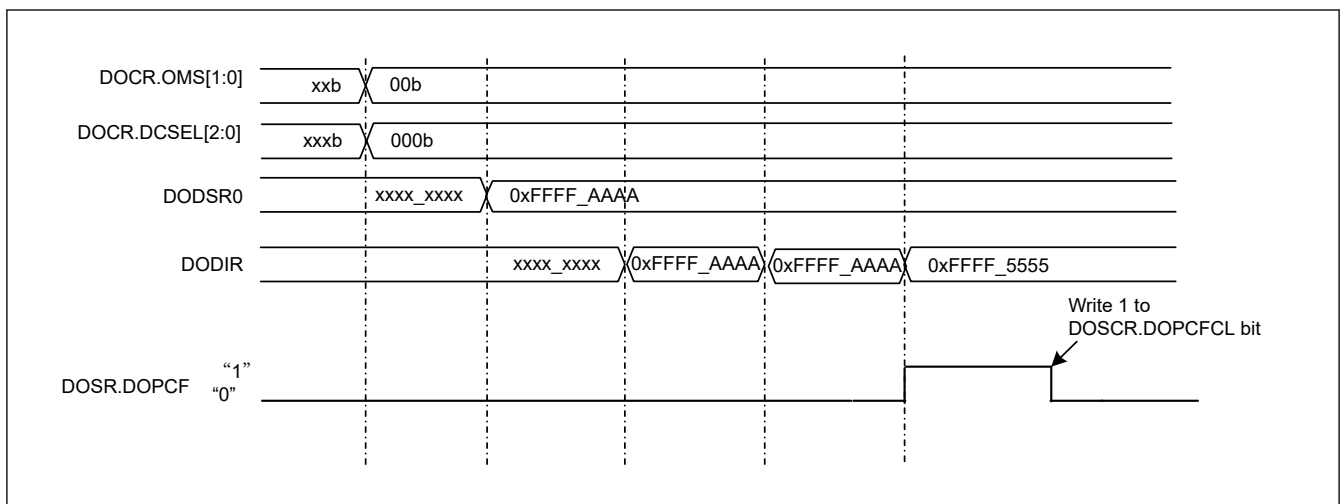


Figure 46.2 Example of Operation in Data Comparison Mode (Detection condition: Mismatch)

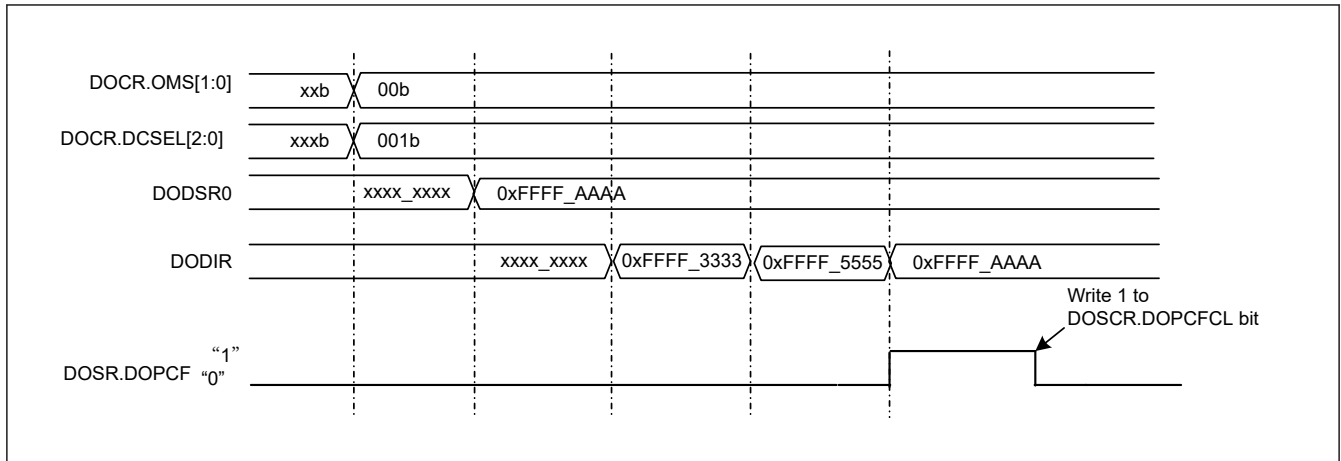


Figure 46.3 Example of Operation in Data Comparison Mode (Detection condition: Match)

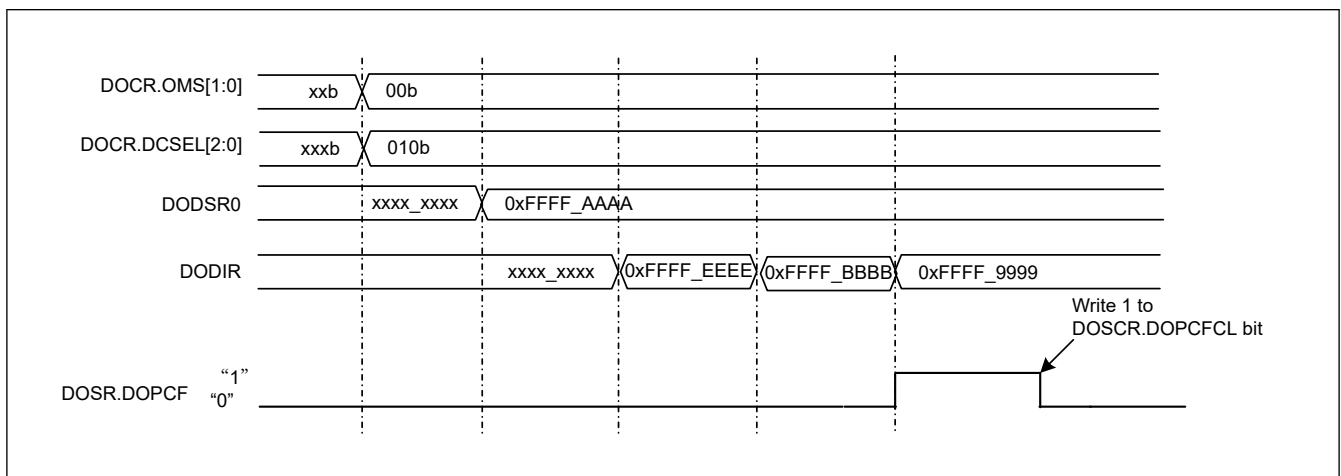


Figure 46.4 Example of Operation in Data Comparison Mode (Detection condition: Lower)

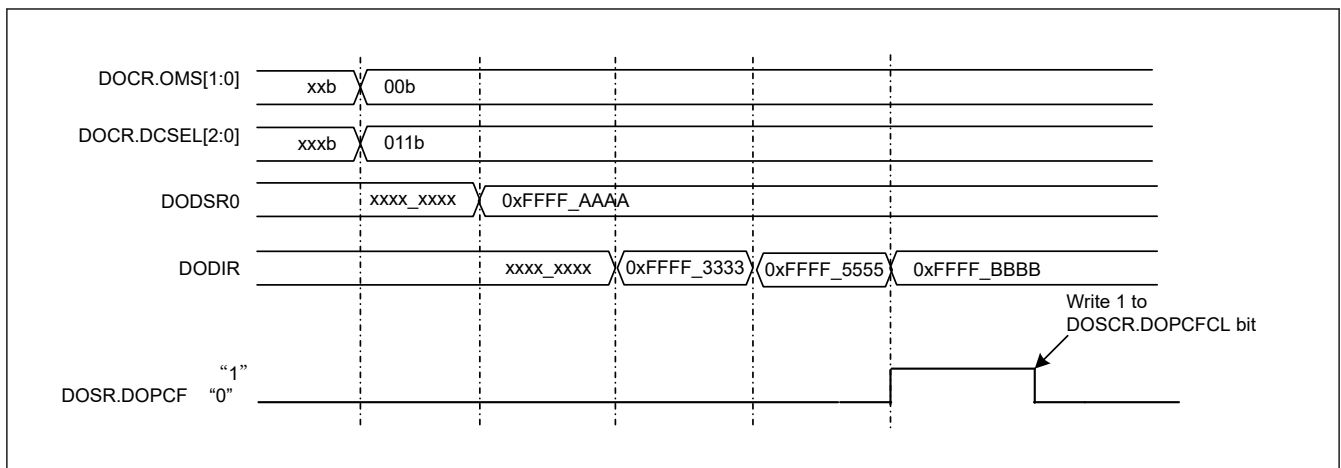


Figure 46.5 Example of Operation in Data Comparison Mode (Detection condition: Upper)

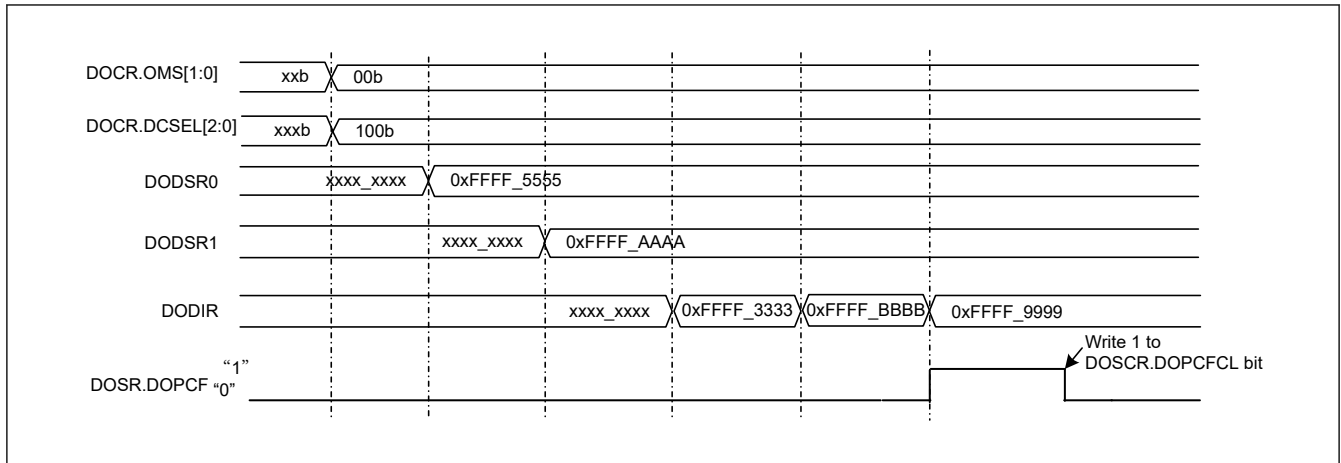


Figure 46.6 Example of Operation in Data Comparison Mode (Detection condition: Inside window)

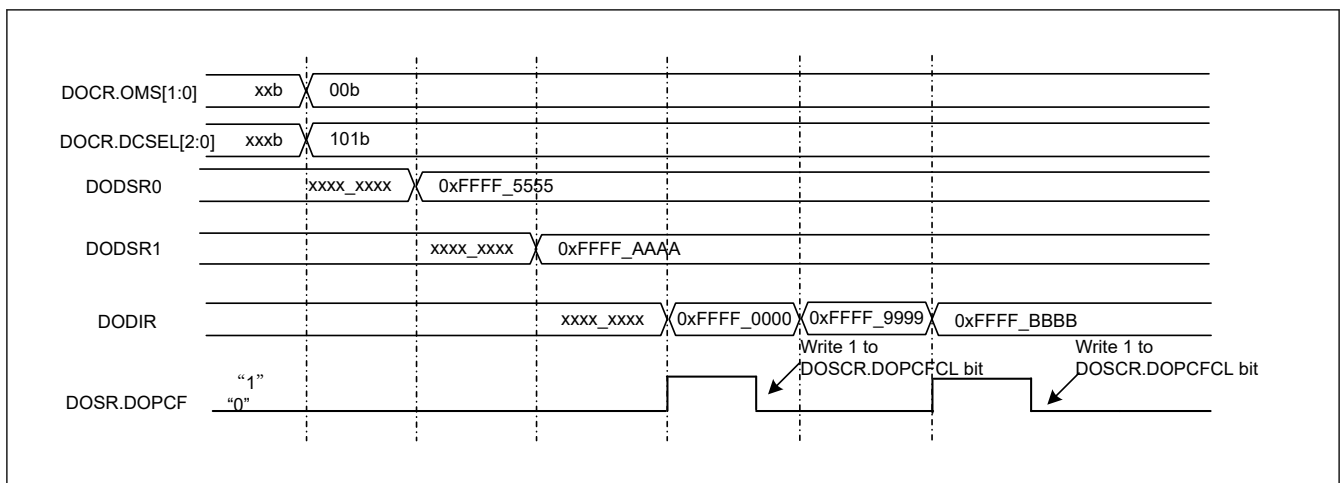


Figure 46.7 Example of Operation in Data Comparison Mode (Detection condition: Outside window)

### 46.3.2 Data Addition Mode

Figure 46.8 shows an example of the steps involved in data addition mode <sup>\*1</sup> operation by the data operation circuit.

The following is an example of operation when the bit width of data operation is 32-bit.

1. Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
2. 32-bit data is set in the DODSR0 register as the initial value.
3. 32-bit data to be added is written to DODIR. The result of the operation is stored in DODSR0.
4. Writing of 32-bit data continues until all data for addition have been written to DODIR.
5. If the result of an operation is greater than 0xFFFF\_FFFF, the DOSR.DOPCF flag is set to 1 and an ELC event and a data operation circuit interrupt are generated.

Note 1. Addition is executed only by writing to the DODIR.

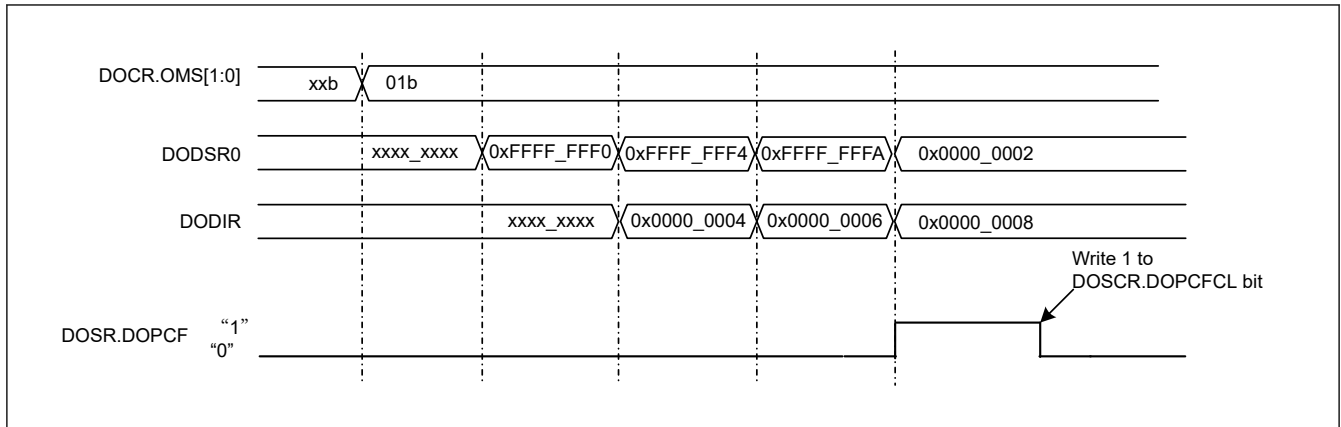


Figure 46.8 Example of Operation in Data Addition Mode

### 46.3.3 Data Subtraction Mode

Figure 46.9 shows an example of the steps involved in data subtraction mode <sup>\*1</sup> operation by the data operation circuit.

The following is an example of operation when the bit width of data operation is 32-bit.

1. Writing 10b to the DOCSR.OMS[1:0] bits selects data subtraction mode.
2. 32-bit data is set in the DODSR0 register as the initial value.
3. 32-bit data to be subtracted is written to DODIR. The result of the operation is stored in DODSR0.
4. Writing of 32-bit data continues until all data for subtraction have been written to DODIR.
5. If the result of an operation is less than 0x0000\_0000, the DOSR.DOPCF flag is set to 1 and an ELC event and a data operation circuit interrupt are generated.

Note 1. Subtraction is executed only by writing to the DODIR.

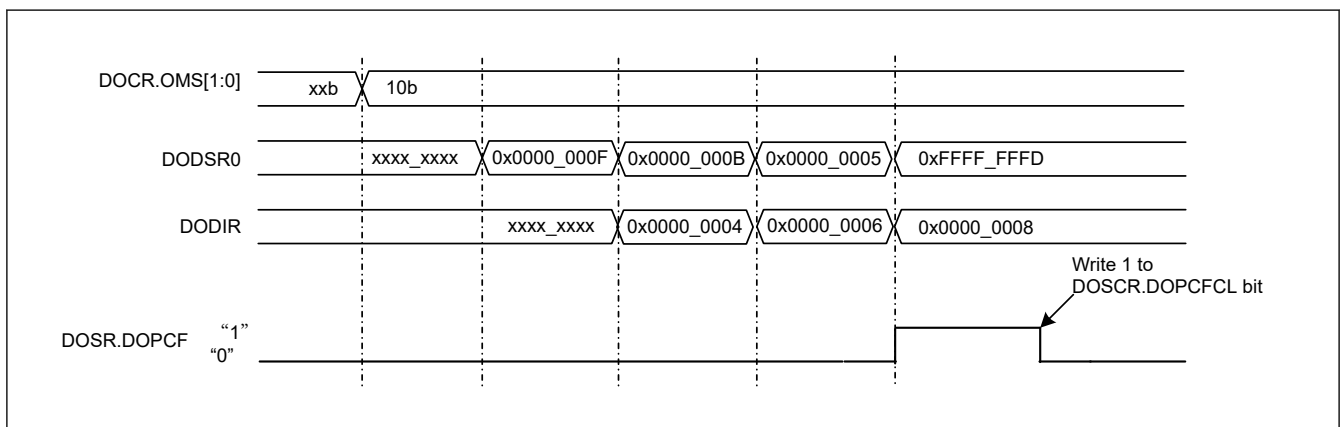


Figure 46.9 Example of Operation in Data Subtraction Mode

## 46.4 Interrupt Source

The data operation circuit generates the data operation circuit interrupt (DOC\_DOPCI) as an interrupt request. When an interrupt source is generated, the data operation circuit flag corresponding to the interrupt is set to 1, and then interrupt request signal is generated. Table 46.2 describes the interrupt request.



**Table 46.2** Interrupt request from DOC

Interrupt request	Status flag	Interrupt source
DOC interrupt	DOPCF	<ul style="list-style-type: none"> <li>The compared values match the detection condition.</li> <li>The result of data addition is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF_FFFF (DOCR.DOBW = 1).</li> <li>The result of data subtraction is less than 0x0000 (DOCR.DOBW = 0) or 0x0000_0000 (DOCR.DOBW = 1).</li> </ul>

## 46.5 Event Link Output

The DOC outputs event signals for the event link controller (ELC) under the following conditions, and these can be used to initiate operations by other modules selected in advance.

- The compared values match the detection condition
- The data addition result is greater than 0xFFFF (DOCR.DOBW = 0) or 0xFFFF\_FFFF (DOCR.DOBW = 1)
- The data subtraction result is less than 0x0000 (DOCR.DOBW = 0) or 0x0000\_0000 (DOCR.DOBW = 1)

## 46.6 Interrupt Handling and Event Linking

The DOC has a bit to enable or disable interrupts. An interrupt request signal is output for the CPU when an interrupt source is generated while the corresponding enable bit is enabled.

In contrast, an event link output signal is sent to other modules as an event signal via the ELC when an interrupt source is generated, regardless of the setting of the corresponding interrupt enable bit.

## 46.7 Usage Notes

### 46.7.1 Settings for the Module-Stop State

The module Stop Control Register C (MSTPCRC) can enable or disable DOC operation. The DOC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

## 47. SRAM

### 47.1 Overview

The MCU provides an on-chip, high-density SRAM module with parity-bit checking.

Table 47.1 lists the SRAM specifications.

**Table 47.1 SRAM specifications**

Parameter	SRAM1
SRAM capacity	512 KB
SRAM address	0x2206_0000 to 0x220D_FFFF (Secure alias), 0x3206_0000 to 0x320D_FFFF (Non-secure alias)
Access	Wait states are inserted into the read cycle by default. If the ICLK frequency is higher than 120 MHz, a wait state is required. If the ICLK frequency is 120 MHz or less, a wait state is not required.
Data retention function	Not available in Deep Software Standby mode
Module-stop function	Module-stop state can be set to reduce power consumption
Error checking	Even-parity (Data: 8 bits, Parity: 1 bit)
Security	TrustZone Filter is integrated for memory access and SFR access. Access to the memory space is controlled by setting the memory Security Attribution (SA). And access to I/O space (SFR) space is controlled by setting the register SA. See <a href="#">section 47.3.3. TrustZone Filter function</a> .

### 47.2 Register Descriptions

#### 47.2.1 SRAMSABAR1 : SRAM Error Address Register

Base address: CPSCU = 0x4000\_8000  
CPSCU\_NS = 0x5000\_8000

Offset address: 0x404

Bit position: 31 0

Bit field: n/a

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Boundary address between Secure and Non-secure. (Start address of Non-secure region)	R/W

Note: S-TYPE-1, P-TYPE-1

Note: This register is write-protected by PRCR\_S.PRC4 register.

SRAMSABAR1 specifies the boundary address between the Secure and Non-secure regions of SRAM. Write the absolute address to the SRAMSABAR1. When writing to SRAMSABAR1, writing from b31 to b21 is ignored and the value written from b12 to b0 should be 0.

The region lower than the boundary address is marked as Secure, and higher than or equal to the boundary address is marked as Non-secure.

The boundary address is as follow.

0x2200\_0000 + SRAMSABAR1 (Secure alias)

0x3200\_0000 + SRAMSABAR1 (Non-secure alias)

When the boundary address is 0x00060000 or less, whole SRAM1 is marked as Non-secure.

When the boundary address is 0x000E0000 or large, whole SRAM1 is marked as Secure.

### 47.2.2 SRAMSAR : SRAM Security Attribution Register

Base address: CPSCU = 0x4000\_8000  
 CPSCU\_NS = 0x5000\_8000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SRAM WTSA	STBR AMSA	—	—	—	—	—	SRAM SA1	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	SRAMSA1	SRAM1 Register Security Attribution 0: Secure 1: Non-secure	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	STBRAMSA	Standby SRAM Register Security Attribution 0: Secure 1: Non-secure	R/W
8	SRAMWTSA	SRAMWTSC Security Attribution 0: Secure 1: Non-secure	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

Note: Only Secure access can write to this register. Both Secure access and Non-secure read access are allowed but Non-secure write access is not allowed and TrustZone access error is not generated.

Note: This register is write-protected by PRCR\_S.PRC4 register.

#### SRAMSA1 bit (SRAM1 Register Security Attribution)

The SRAMSA1 bit specifies security attributes of SRAMCR1 and SRAMESCLR.CLR1.

#### STBRAMSA bit (Standby SRAM Register Security Attribution)

The STBRAMSA bit specifies security attributes of STBRAMCR and SRAMESCLR.CLRS.

#### SRAMWTSA bit (SRAMWTSC Security Attribution)

The SRAMWTSA bit specifies security attributes of SRAMWTSC.

### 47.2.3 SRAMPRCR\_S : SRAM Protection Control Register for Secure

Base address: SRAM = 0x4000\_2000

Offset address: 0x00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KW[7:0]							—	—	—	—	—	—	—	—	PR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PR	Register Write Control 0: Writing to registers are disabled 1: Writing to registers are enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KW[7:0]	Write Key Code Key code protection to the PR bit	R/W

Note: S-TYPE-6, P-TYPE-2

Note: It is necessary to write by half word access.

Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

### PR bit (Register Write Control)

The PR bit controls the write mode of the SRAMWTSC, SRAMCR1, and STBRAMCR registers that are marked as Secure by SRAMSAR register. When this bit is set to 1, writing to the SRAMWTSC, SRAMCR1, and STBRAMCR marked as Secure is enabled.

While writing to this register it is necessary to write 0xA5 to the KW[7:0] bits simultaneously.

### KW[7:0] bits (Write Key Code)

The KW[7:0] bits enable or disable writing to the PR bit. When you write to the PR bit, write 0xA5 to the KW[7:0] bits simultaneously. When a value other than 0xA5 is written to KW[7:0], the PR bit is not updated. The KW[7:0] bits are always read as 0x00.

## 47.2.4 SRAMPRCR\_NS : SRAM Protection Control Register for Non-secure

Base address: SRAM\_NS = 0x5000\_2000

Offset address: 0x04

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	KW[7:0]							—	—	—	—	—	—	—	PR
------------	---------	--	--	--	--	--	--	---	---	---	---	---	---	---	----

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	PR	Register Write Control 0: Writing to registers are disabled 1: Writing to registers are enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KW[7:0]	Write Key Code Key code protection to the PR bit	R/W

Note: S-TYPE-7, P-TYPE-2

Note: It is necessary to write by half word access.

Byte write access is prohibited. When byte write access is executed, operation is not guaranteed.

### PR bit (Register Write Control)

The PR bit controls the write mode of the SRAMWTSC, SRAMCR1, and STBRAMCR registers that are marked as Non-secure by SRAMSAR register. When this bit is set to 1, writing to the SRAMWTSC, SRAMCR1, and STBRAMCR marked as Non-secure is enabled.

While writing to this register it is necessary to write 0xA5 to the KW[7:0] bits simultaneously.

### KW[7:0] bits (Write Key Code)

The KW[7:0] bits enable or disable writing to the PR bit. When you write to the PR bit, write 0xA5 to the KW[7:0] bits simultaneously. When a value other than 0xA5 is written to KW[7:0], the PR bit is not updated. The KW[7:0] bits are always read as 0x00.

### 47.2.5 SRAMWTSC : SRAM Wait State Control Register

Base address: SRAM = 0x4000\_2000  
SRAM\_NS = 0x5000\_2000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	WTEN

Value after reset: 0 0 0 0 0 0 0 0 1

Bit	Symbol	Function	R/W
0	WTEN	SRAM Wait Enable 0: No wait 1: Add wait state in read access cycle to SRAMs	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (SRAMSAR.SRAMWTSA), P-TYPE-2

This register can be written only when the PR bit in the SRAMPRCR\_S or SRAMPRCR\_NS register is 1.

#### WTEN bit (SRAM Wait Enable)

The WTEN bit sets the wait cycle insertion to the access cycle of SRAM. When 1 is set to the WTEN bit, 1 wait cycle is inserted to the read access cycle of SRAM. Also when the WTEN bit is 1, the access to the same region<sup>\*1</sup> and the same mat<sup>\*2</sup> is continuous and 1 wait cycle is inserted to the second access cycle. When the access frequency is more than 120 MHz, it is necessary to set 1 wait cycle to the WTEN bit.

Note 1. The region of SRAM is divided into 128 KB units.

Note 2. The mat is divided into 0x0 to 0x7 and 0x8 to 0xF in the lower 4 bits of the address.

For example, the same region and the same mat is 8 areas of SRAM1 in the figure below. If 1 area is accessed continuously, 1 wait cycle is inserted.

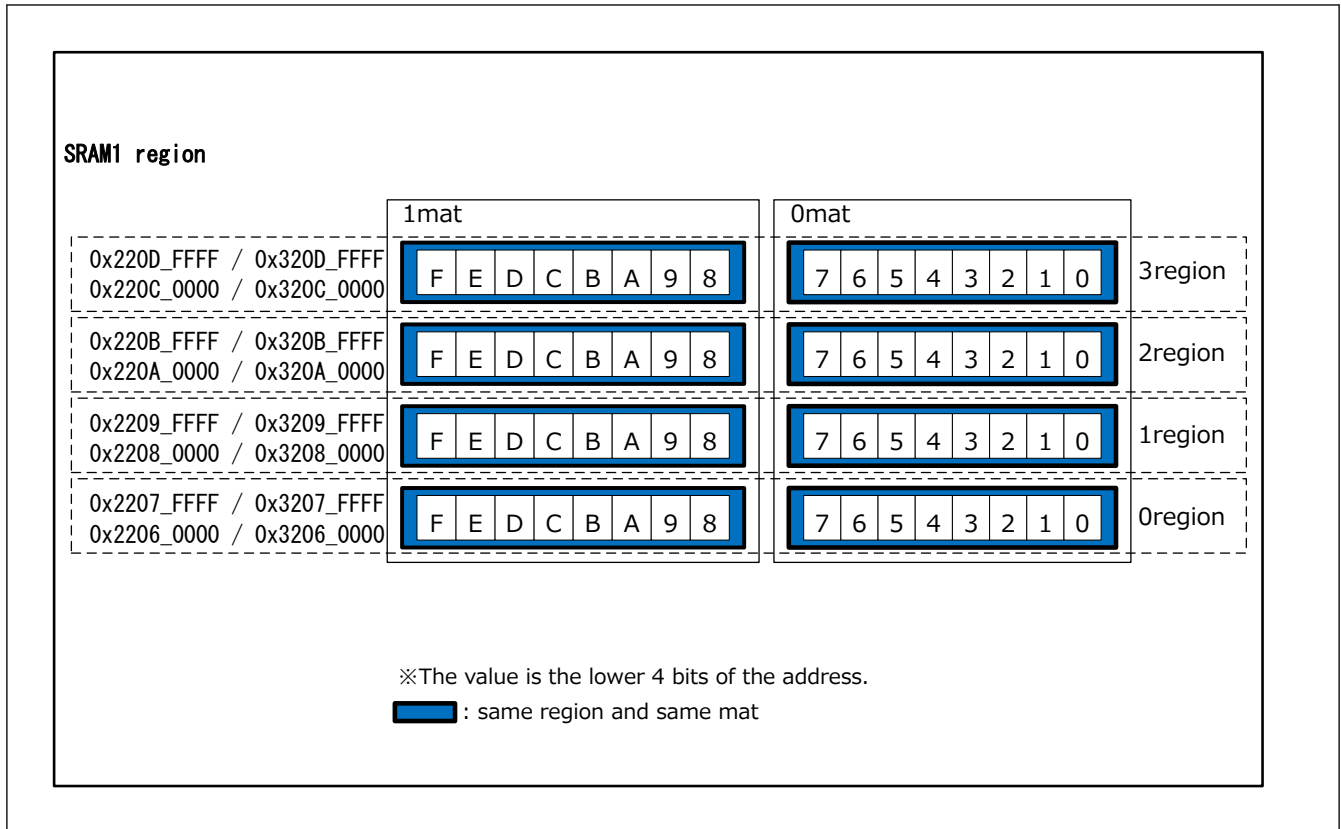


Figure 47.1 Mat configuration of SRAM

### 47.2.6 SRAMCR1 : SRAM Control Register 1

Base address: SRAM = 0x4000\_2000  
 SRAM\_NS = 0x5000\_2000

Offset address: 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	OAD

Value after reset: 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	OAD	Operation after Detection for Parity Error Detection 0: Non-maskable interrupt 1: Reset.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (SRAMSAR.SRAMSA1), P-TYPE-2

This register can be written only when the PR bit in the SRAMPRCR\_S or SRAMPRCR\_NS register is 1.

Do not write to this register while access to SRAM is in progress. See [section 47.4.2. Note of write to SRAMCR1 register](#) for writing to this register.

#### OAD bit (Operation after Detection for Parity Error Detection)

The OAD bit selects a reset or non-maskable interrupt when a parity error is detected.

### 47.2.7 SRAMESR : SRAM Error Status Register

Base address: SRAM = 0x4000\_2000  
SRAM\_NS = 0x5000\_2000

Offset address: 0x40

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ERRS	—	—	—	—	—	—	—	—	—	—	—	ERR1	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0.	R
2	ERR1	SRAM1 Parity Error Status 0: Parity error has not occurred. 1: Parity error has occurred.	R
13:3	—	These bits are read as 0.	R
14	ERRS	Standby SRAM Parity Error status 0: Parity error has not occurred. 1: Parity error has occurred.	R
15	—	This bit is read as 0.	R

Note: S-TYPE-5, P-TYPE-2

This register is cleared by the corresponding bit in the SRAMESCLR register or resets other than Bus Error Reset and Memory Error Reset. If error and clear occur at the same time, clear has priority. Also updating register stops during accessing from debugger.

#### ERR1 bit (SRAM1 Parity Error Status)

The ERR1 bit shows whether there is a parity error or not in the SRAM1.

The ERR1 bit is set if a parity error is detected.

When the ERR1 bit is set to 1, a reset or non-maskable interrupt request is generated according to the SRAMCR1.OAD.

#### ERRS bit (Standby SRAM Parity Error status)

The ERRS bit shows whether there is a parity error or not in the Standby SRAM.

The ERRS bit is set to 1 if there is an error.

When the ERRS bit is set to 1, a reset or non-maskable interrupt request is generated according to the STBRAMCR.OAD.

### 47.2.8 SRAMESCLR : SRAM Error Status Clear Register

Base address: SRAM = 0x4000\_2000  
SRAM\_NS = 0x5000\_2000

Offset address: 0x48

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CLRS	—	—	—	—	—	—	—	—	—	—	—	—	CLR1	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W <sup>1</sup>
2	CLR1	SRAM1 Parity Error Status Clear Writing to the CLR1 clears SRAMESR.ERR1. 1: Clear Parity error. Read value is always 0.	R/W <sup>1</sup>

Bit	Symbol	Function	R/W
13:3	—	These bits are read as 0. The write value should be 0.	R/W
14	CLRS	Standby SRAM Parity Error Status Clear Writing to the CLRS clears SRAMESR. ERRS. 1: Clear Parity error. Read value is always 0.	R/W <sup>1</sup>
15	—	This bit is read as 0. The write value should be 0.	R/W

Note: S-TYPE-4 (SRAMSAR.SRAMSA1, STBRAMSA), P-TYPE-2

Note 1. For each bit, only 1 can be written to clear the error status.

### CLR1 bit (SRAM1 Parity Error Status Clear)

The CLR1 bit can clear the SRAM1 parity error status bit in the SRAMESR. If error and clear occur at the same time, clear has priority.

The SRAMESR.ERR1 bit can be cleared when writing 1 to CLR1.

### CLRS bits (Standby SRAM Parity Error Status Clear)

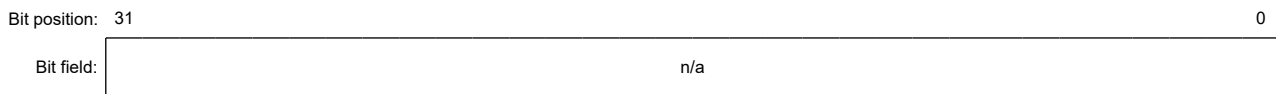
The CLRS bits can clear the Standby SRAM Parity error bit in the SRAMESR. If error and clear occur at the same time, clear has priority.

The SRAMESR.ERRS bit can be cleared when writing 1 to CLRS.

## 47.2.9 SRAMEAR2 : SRAM Error Address Register

Base address: SRAM = 0x4000\_2000  
SRAM\_NS = 0x5000\_2000

Offset address: 0x58



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	When an SRAM error occurs, it stores an error address.	R

Note: S-TYPE-5, P-TYPE-2

This register is cleared by the corresponding bit in the SRAMESCLR register or resets other than Bus Error Reset and Memory Error Reset. If error and clear occur at the same time, clear has priority. Also updating register stops during accessing from debugger.

The error address is as follow.

0x2200\_0000 + SRAMEAR2 (Secure alias)

0x3200\_0000 + SRAMEAR2 (Non-secure alias)

This register stores the error address where parity error is detected. These bits hold the error address that occurred first.

## 47.3 Operation

### 47.3.1 Module Stop Function

Power consumption can be reduced by setting module stop control register A (MSTPCRA) to stop supply of the clock signal to SRAM.

SRAM1 is controlled by MSTPA1 bit in MSTPCRA register and, in the case of 1, SRAM1 becomes the clock stop state.

The SRAM is thus placed in the module-stop state by stopping supply of the clock signals. The SRAM operates after a reset.

SRAM is not accessible if it is in the module-stop state. A transition to the module-stop state should not be made while access to SRAM is in progress.



Access to the SRAM in the module-stop state is prohibited. If access is attempted, correct operation is not guaranteed. For details on the MSTPCRA register, see [section 10, Low Power Modes](#).

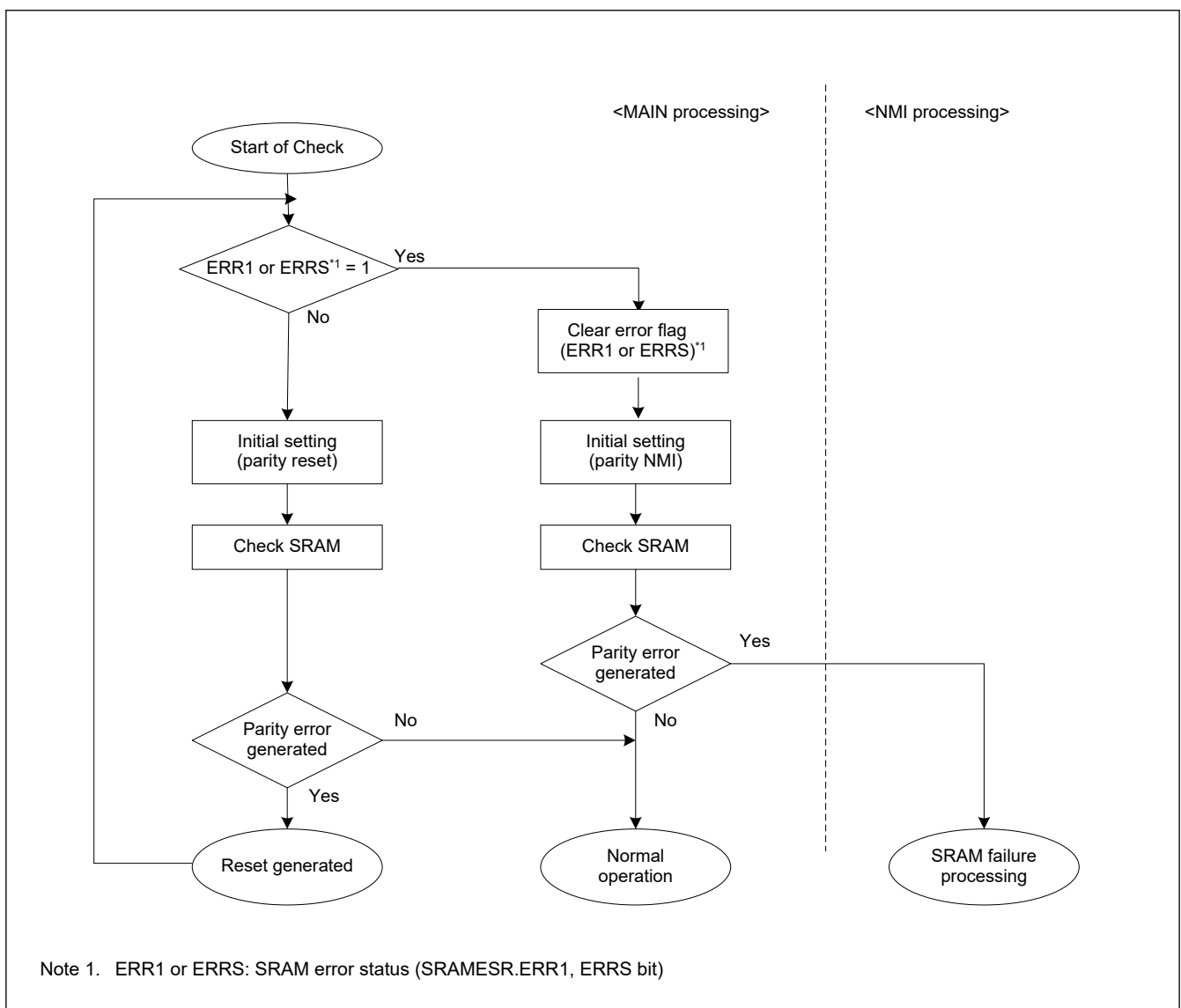
### 47.3.2 Parity Calculation Function

The IEC60730 standard requires the checking of SRAM data. When data is written, a parity bit is added to every 8-bit data in the SRAM, and when data is read, the parity is checked. When a parity error occurs, a parity-error notification is generated. This function can also be used to trigger a reset.

The parity-error notification can be specified as a non-maskable interrupt or a reset in the OAD bit of the SRAMCR1 register. When the OAD bit is set to 1, a parity error is output to the reset function. When the OAD bit is set to 0, a parity error is output to the ICU as a non-maskable interrupt.

Parity Error often occurs due to noise. It is possible to confirm whether the parity error is caused by noise or destruction of data by following the parity operation flowchart. [Figure 47.2](#) and [Figure 47.3](#) shows flowcharts of SRAM parity check operation..

When a read access is executed in a row after a write access, read access is executed with priority. Therefore, during initialization, do not perform the read access in a row after the write access.



**Figure 47.2** Flow of SRAM parity check when SRAM parity reset is enabled

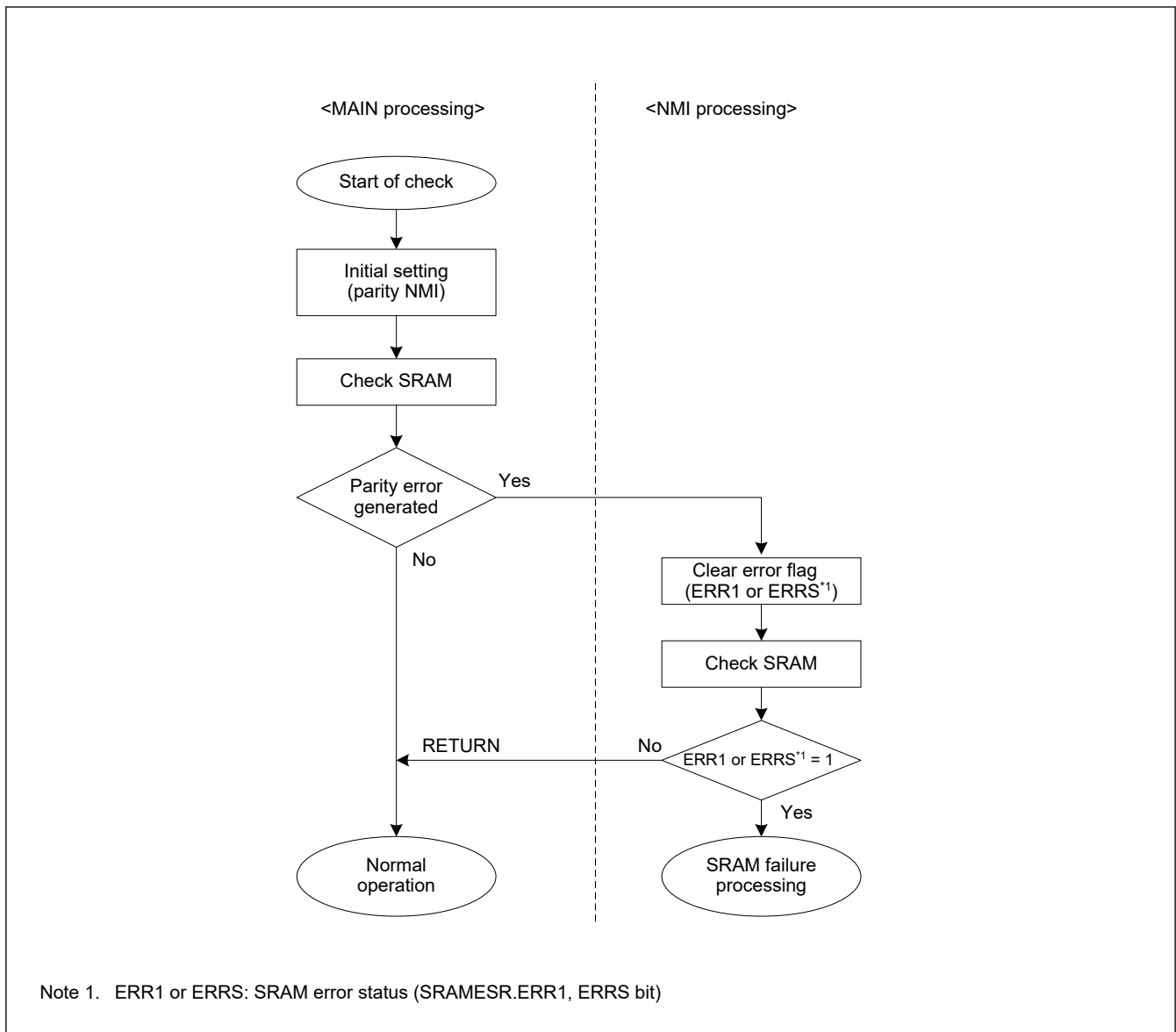


Figure 47.3 Flow of SRAM parity check when SRAM parity interrupt is enabled

### 47.3.3 TrustZone Filter function

#### 47.3.3.1 TrustZone Filter for SRAM Registers

SRAM registers can be protected with a Security Attribution (SA) from Non-secure or Secure transaction. When SA indicates that SRAM registers are Secure status, Non-secure transaction cannot access them because TrustZone Filter detects an error and protects the access. When SA indicates that SRAM registers are Non-secure status, Secure transaction cannot access them because Trust Zone Filter detects an error and protects the access. SA for SRAM registers is just one to be used commonly among SRAM registers.

In the case of a failed access due to TrustZone error, error response is generated.

Table 47.2 Register protection (1 of 2)

SA	Transaction	Write access	Read access
Secure	Secure	Permitted	Permitted
	Non-secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)

**Table 47.2 Register protection (2 of 2)**

SA	Transaction	Write access	Read access
Non-secure	Secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
	Non-secure	Permitted	Permitted

### 47.3.3.2 TrustZone Filter for SRAM Memory Regions

SRAM1 region can be divided into Secure and Non-secure independently. The access permissions for these regions are as below.

**Table 47.3 Memory protection**

SA	Transaction	Write access	Read access
Secure	Secure	Permitted	Permit
	Non-secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
Non-secure	Secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
	Non-secure	Permitted	Permitted

TrustZone filter error for SRAM memory generates an error notification.

### 47.3.4 Interrupt Source

The SRAM interrupt source includes a Parity error and TrustZone filter error. Parity error can choose non-maskable interrupt or reset by SRAMCR1.OAD bit. The SRAM interrupt occurs when one error status in the SRAMESR register is set to 1, the SRAM interrupt continues to occur until the SRAMESR register flag is cleared. When Common memory error occur (NMISR.CMST=1 or RSTSR1.CMSR=1), please read SRAMESR and check SRAM interrupt source. When the access from debugger, the error is detected and corrected, but no error flag is set, reset and non-maskable interrupt are maskable. For details on the debug mode, see [section 2, CPU](#).

**Table 47.4 SRAM Interrupt Source**

Name	Interrupt Source	DTC Activation	DMAC Activation
PARITYERR	Parity error (SRAMs with parity)	Not possible	Not possible
TZFLT	TrustZone filter error	Not possible	Not possible

### 47.3.5 Wait state

When ICLK frequencies is higher than 120 MHz, do not set 0x00 in wait enable bit for the SRAMWTSC register, in order to insert a wait cycle. When the wait is not inserted, the operation is not guaranteed.

Depending on the operating frequency of ICLK, the WAIT setting for SRAM access has the following conditions.

[ICLK frequency]:

- 240 MHz ≥ ICLK > 120 MHz = 1 wait
- 120 MHz ≥ ICLK = No-wait

## 47.4 Usage Note

### 47.4.1 Instruction fetch from SRAM area

When using SRAM to operate a program, initialize the SRAM area so that the CPU can correctly prefetch data. A parity error might occur if the CPU prefetches from an area that is not initialized. Initialize the additional 12-byte area from the end address of the program with the 8-byte boundary. Renesas recommends using the NOP instruction for data initialization.

### 47.4.2 Note of write to SRAMCR1 register

This chapter provides examples of how to configure SRAMCR1.

All of the followings have to be satisfied:

1. CPU does not execute program stored in SRAM
2. All bus masters other than CPU does not access SRAM
3. SRAMCR1 is Device-nGnRnE set by MPU
4. SRAMCR1 have to be configured with the following procedure:
  1. DMB instruction
  2. Write to SRAMCR1 register
  3. DMB instruction

#### 47.4.3 Notes on Using Error Checking of SRAM

Data in SRAM are undefined when the power is turned on. Therefore, parity check errors occur if the data are read before initialization. The SRAM are read in 8-byte (64-bit) units. Initialize them on 8-byte boundaries.

## 48. Standby SRAM

### 48.1 Overview

An on-chip SRAM is provided to retain data in Deep Software Standby mode. [Table 48.1](#) lists the Standby SRAM specifications.

**Table 48.1 Standby SRAM specifications**

Item	Description
SRAM capacity	1 KB
SRAM address	0x2600_0000 to 0x2600_03FF (Secure alias), 0x3600_0000 to 0x3600_03FF (Non-secure alias)
Access	Wait states are inserted into the access cycle by default. If the ICLK frequency is higher than 120 MHz, a wait state is required. If the ICLK frequency is 120 MHz or less, a wait state is not required.
Data retention function	Data can be retained in Deep Software Standby mode 1. In Deep Software Standby mode 2 and 3, data cannot be retained. See <a href="#">section 48.3.1. Data Retention</a> for details.
parity	Even parity (data: 8 bits, parity: 1 bit)
Module-stop function	Module-stop state can be set to reduce power consumption. See <a href="#">section 48.3.2. Module-stop Function</a> for details.
Security	Permits the read and write operations to Standby RAM following TrustZone Filter function. See <a href="#">section 48.3.4. TrustZone Filter function</a> for details.

### 48.2 Register Descriptions

#### 48.2.1 STBRAMSABAR : Standby SRAM Security Attribute Boundary Address Register

Base address: CPSCU = 0x4000\_8000  
CPSCU\_NS = 0x5000\_8000

Offset address: 0x420

Bit position: 31 0

Bit field:

Value after reset: 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
31:0	n/a	Boundary address between Secure and Non-secure.	R/W

Note: S-TYPE-1, P-TYPE-1

Note: This register is write-protected by PRCR\_S.PRC4 register.

STBRAMSABAR specifies the boundary address between the Secure and Non-secure regions of Standby SRAM. Write the absolute address to the STBRAMSABAR. When writing to STBRAMSABAR, writing from b31 to b15 is ignored and the value written from b6 to b0 should be 0.

The region lower than the boundary address is marked as Secure, and higher than or equal to the boundary address is marked as Non-secure. Therefore, when the boundary address is 0x00000000, whole Standby SRAM is marked as Non-secure, and when the boundary address is 0x00000400 or large, whole Standby SRAM is marked as Secure.

The boundary address is as follow.

- 0x2600\_0000 + STBRAMSABAR (Secure alias)
- 0x3600\_0000 + STBRAMSABAR (Non-secure alias)

### 48.2.2 STBRAMPABAR\_NS : Standby SRAM Privilege Attribute Boundary Address Register for Non-secure

Base address: CPSCU\_NS = 0x5000\_8000

Offset address: 0x490

Bit position: 31 0

Bit field: n/a

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	Boundary address between Privileged and Unprivileged.	R/W

Note: S-TYPE-7, P-TYPE-2

Note: This register is write-protected by PRCR\_NS.PRC4 register.

STBRAMPABAR\_NS specifies the boundary address between the Privileged and Unprivileged regions of Standby SRAM Non-secure regions set by STBRAMSABAR. Write the absolute address to the STBRAMPABAR\_NS. When writing to STBRAMPABAR\_NS, writing from b31 to b15 is ignored and the value written from b6 to b0 should be 0.

The region lower than the boundary address is marked as Privileged, and higher than or equal to the boundary address is marked as Unprivileged. Therefore, when the boundary address is 0x00000000, or less than or equal to start address of Standby SRAM Non-secure regions set by STBRAMSABAR, whole Standby SRAM is marked as Unprivileged, and when the boundary address is 0x00000400 or large, or higher than or equal to end address of Standby SRAM Non-secure regions set by STBRAMSABAR, whole Standby SRAM is marked as Privileged.

The boundary address is as follow.

- 0x3600\_0000 + STBRAMPABAR\_NS (Non-secure alias)

### 48.2.3 STBRAMPABAR\_S : Standby SRAM Privilege Attribute Boundary Address Register for Secure

Base address: CPSCU = 0x4000\_8000

Offset address: 0x494

Bit position: 31 0

Bit field: n/a

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	n/a	Boundary address between Privileged and Unprivileged.	R/W

Note: S-TYPE-6, P-TYPE-2

Note: This register is write-protected by PRCR\_S.PRC4 register.

STBRAMPABAR\_S specifies the boundary address between the Privileged and Unprivileged regions of Standby SRAM Secure regions set by STBRAMSABAR. Write the absolute address to the STBRAMPABAR\_S. When writing to STBRAMPABAR\_S, writing from b31 to b15 is ignored and the value written from b6 to b0 should be 0.

The region lower than the boundary address is marked as Privileged, and higher than or equal to the boundary address is marked as Unprivileged. Therefore, when the boundary address is 0x00000000, or less than or equal to start address of Standby SRAM Secure regions set by STBRAMSABAR, whole Standby SRAM is marked as Unprivileged, and when the boundary address is 0x00000400 or large, or higher than or equal to end address of Standby SRAM Secure regions set by STBRAMSABAR, whole Standby SRAM is marked as Privileged.

The boundary address is as follow.

- 0x2600\_0000 + STBRAMPABAR\_S (Secure alias)

## 48.2.4 STBRAMCR : Standby SRAM Control Register

Base address: SRAM = 0x4000\_2000  
SRAM\_NS = 0x5000\_2000

Offset address: 0x110

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	OAD
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OAD	Operation after detection 0: Non-maskable interrupt. 1: Reset.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3 (SRAMSAR.STBRAMSA), P-TYPE-2

This register can be written only when the PR bit in the SRAMPRCR\_S or SRAMPRCR\_NS register is 1.

Do not write to this register while access to Standby SRAM is in progress.

### OAD bit (Operation after detection)

This bit is specified to generate either reset or non-maskable interrupt when parity error is detected.

## 48.2.5 STBRAMEAR : Standby SRAM Error Address Register

Base address: SRAM = 0x4000\_2000  
SRAM\_NS = 0x5000\_2000

Offset address: 0x150

Bit position:	31		0
Bit field:	n/a		
Value after reset:	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	When an SRAM error occurs, it stores an error address.	R

Note: S-TYPE-5, P-TYPE-2

This register is cleared by the corresponding bit in SRAMESCLR register, or resets other than Bus Error Reset and Memory Error Reset. If error and clear occur at the same time, clear has priority. Also during accessing from debugger it is to stop updating register.

The error address is as follow.

- 0x2600\_0000 + STBRAMEAR (Secure alias)
- 0x3600\_0000 + STBRAMEAR (Non-secure alias)

This register stores the error address where parity error is detected. These bits hold the error address that occurred first. These bits are cleared by clearing parity error from SRAMESCLR.

## 48.3 Operation

### 48.3.1 Data Retention

The power supply to the Standby SRAM in Deep Software Standby mode is enabled by the DPSBYCR.SRKEEP bit. If the DPSBYCR.SRKEEP bit are set to 1b, data in the Standby SRAM is retained in Deep Software Standby mode 1. For details on the DPSBYCR.SRKEEP bit, see [section 10, Low Power Modes](#).

### 48.3.2 Module-stop Function

Power consumption can be reduced by setting module stop control register A (MSTPCRA) to stop supply of the clock signal to SRAM.

If the Standby SRAM bit in MSTPCRA is set to 1, supply of the clock signal to the Standby SRAM is stopped.

The Standby SRAM is thus placed in the module-stop state by stopping supply of the clock signals. The Standby SRAM operates after a reset.

The Standby SRAM is not accessible if it is in the module-stop state. A transition to the module-stop state should not be made while access to the standby SRAM is in progress.

Access to the Standby SRAM in the module-stop state is prohibited. If access is attempted, correct operation is not guaranteed.

For details on the MSTPCRA register, see [section 10, Low Power Modes](#).

### 48.3.3 Parity Calculation Function

The IEC60730 standard requires the checking of STBRAMCR data. When data is written, a parity bit is added to every 8-bit data in the Standby SRAM which has 32-bit data width, and when data is read, the parity is checked. When a parity error occurs, a parity error notification is generated. This function can also be used to trigger a reset.

The parity-error notification can be specified as a non-maskable interrupt or a reset in the OAD bit of the STBRAMCR register. When the OAD bit is set to 1, a parity error is output to the reset function. When the OAD bit is set to 0, a parity error is output to the ICU as a non-maskable interrupt.

Parity errors often occur because of noise. To confirm whether the cause of the parity error is noise or corruption, follow the parity check flows shown in [Figure 48.1](#) and [Figure 48.2](#).

When a read access is executed in a row after a write access, read access is executed with priority. Therefore, during initialization, do not perform the read access in a row after the write access.



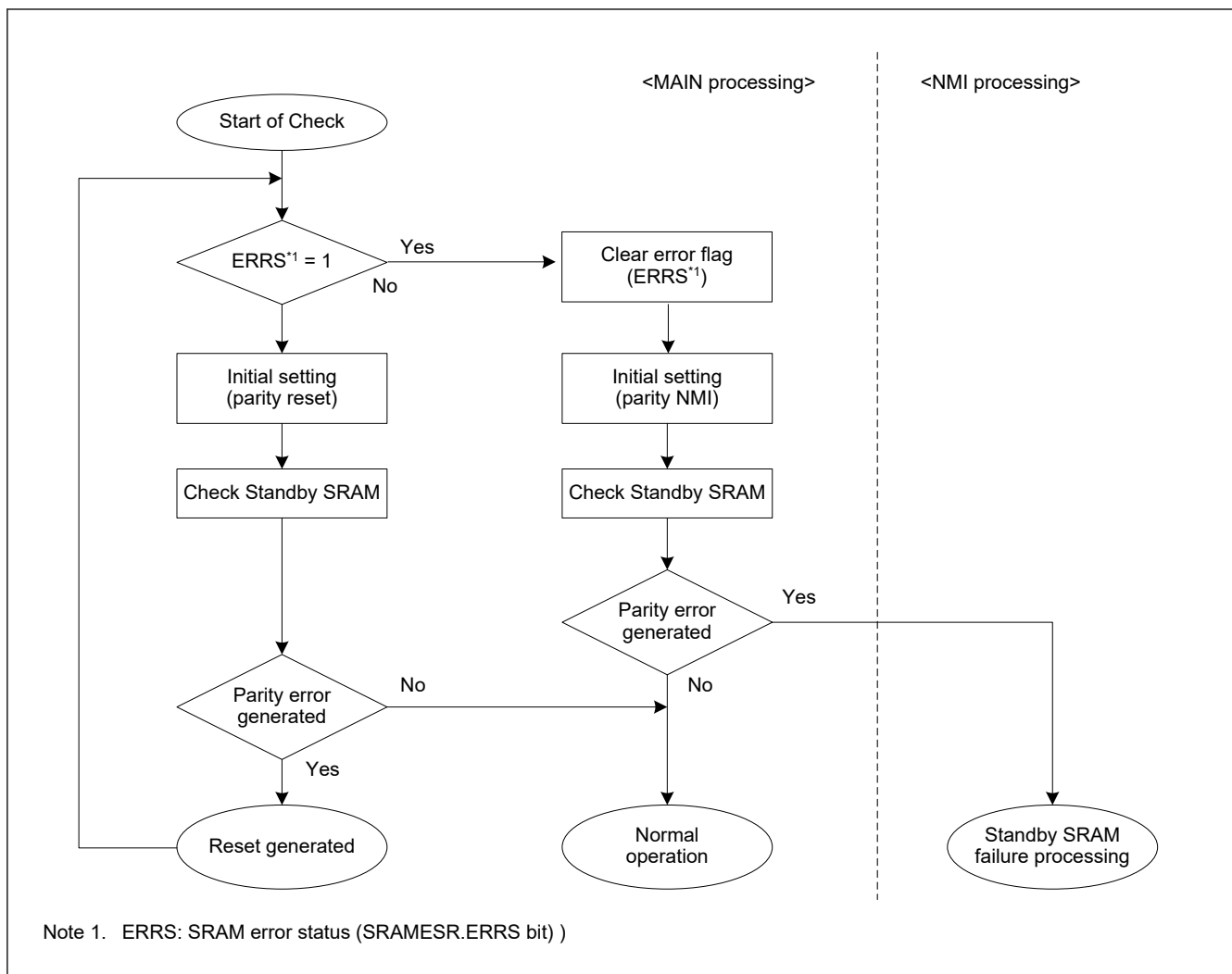


Figure 48.1 Flow of Standby SRAM parity check when SRAM parity reset is enabled

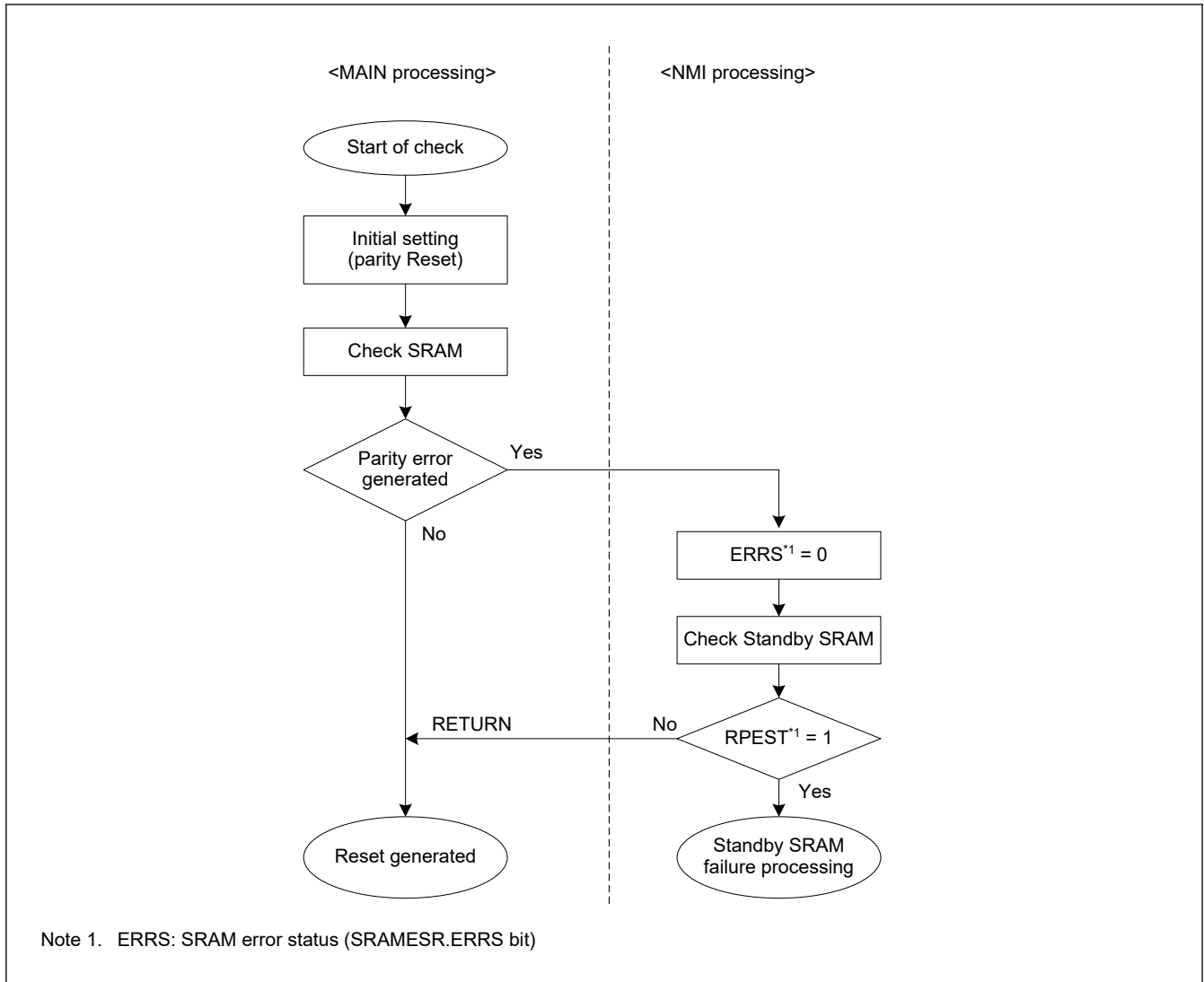


Figure 48.2 Flow of Standby SRAM parity check when SRAM parity interrupt is enabled

### 48.3.4 TrustZone Filter function

#### 48.3.4.1 TrustZone Filter for Standby SRAM registers

Standby SRAM registers can be protected with a Security Attribution (SA) from Non-secure or Secure transaction. When SA indicates that Standby SRAM registers are Secure status, Non-secure transaction can not access them because TrustZone Filter detects finds an error and protects the access. When SA indicates that SRAM registers are Non-secure status, Secure transaction can not access them because Trust Zone Filter detects finds an error and protects the access. SA for Standby SRAM registers is just one to be used commonly among Standby SRAM registers.

In the case of a failed access due to TrustZone error, error response is generated.

Table 48.2 Standby SRAM register protection

SA	Transaction	Write access	Read access
Secure	Secure	Permitted	Permitted
	Non-secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
Non-secure	Secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
	Non-secure	Permitted	Permitted

#### 48.3.4.2 TrustZone Filter for Standby SRAM memory regions

Standby SRAM regions can be divided into Secure and Non-secure. The access permissions for these regions are as below.

**Table 48.3 Standby SRAM memory protection by Security Attribution**

SA	Transaction	Write access	Read access
Secure	Secure	Permitted	Permitted
	Non-secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
Non-secure	Secure	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
	Non-secure	Permitted	Permitted

Standby SRAM regions can be divided into Privileged and Unprivileged. The access permissions for these regions are as below.

**Table 48.4 Standby SRAM memory protection by Privilege Attribution**

PA	Transaction	Write access	Read access
Privileged	Privileged	Permitted	Permitted
	Unprivileged	Protected (TrustZone Filter error)	Protected (TrustZone Filter error)
Unprivileged	Privileged	Permitted	Permitted
	Unprivileged	Permitted	Permitted

TrustZone Filter error for Standby SRAM memory generates an error notification.

For details, see [section 40, Security Features](#) .

#### 48.3.5 Interrupt Source

The Standby SRAM interrupt source includes a Parity error and TrustZone filter error.

Parity error can choose non-maskable interrupt or reset by STBRAMCR.OAD bit.

The Standby SRAM interrupt occurs when the SRAMESR.ERRS is set to 1. the Standby SRAM interrupt continues to occur until the SRAMESR.ERRS is cleared.

When Common memory error occur(NMISR.CMST=1 or RSTSR1.CMSR=1), please read SRAMESR and check SRAM interrupt source. When the access from debugger, no error flag is set, reset and non-maskable interrupt are maskable.

For more detail on the debug mode, see [section 2, CPU](#).

**Table 48.5 Standby SRAM Interrupt Source**

Name	Interrupt Source	DTC Activation	DMAC Activation
PARITYERR	Parity error	Not possible	Not possible
TZFLT	TrustZone filter error	Not possible	Not possible

#### 48.3.6 Wait state

When ICLK frequencies is higher than 120 MHz, do not set 0x00 in wait enable bit for the SRAMWTSC register, in order to insert a wait cycle. When the wait is not inserted, the operation is not guaranteed.

Depending on the operating frequency of ICLK, the WAIT setting for Standby SRAM access has the following conditions.

[ICLK frequency]:

- $240 \text{ MHz} \geq \text{ICLK} > 120 \text{ MHz} = 1 \text{ wait}$
- $120 \text{ MHz} \geq \text{ICLK} = \text{No-wait}$

## 48.4 Usage Notes

### 48.4.1 Instruction Fetch from the Standby SRAM Area

When using Standby SRAM to operate a program, initialize the Standby SRAM area so that the CPU can correctly prefetch data. A parity error might occur if the CPU prefetches from an area that is not initialized. Initialize the additional 12-byte area from the end address of the program with the 4-byte boundary. Renesas recommends using the NOP instruction for data initialization.

### 48.4.2 Notes on Self-Diagnosis of the Standby SRAM

A write buffer is mounted for the Standby Standby SRAM. When the same address is read after a write operation, data in the write buffer, rather than in the memory cell of the Standby SRAM may be read. When the Standby SRAM is self-diagnosed, confirm that the data have been written by following the procedure below so that data will not be read from the write buffer.

1. Write data to the address targeted for diagnosis.
2. Write data to an address which is at least 4 addresses away from the that in step1.
3. Read the data from the address in step1.

## 49. Flash Memory

This MCU incorporates code flash memory, data flash memory, and option-setting memory. The code flash memory stores instructions and operands, and the data flash memory stores data. For option-setting memory, see [section 6, Option-Setting Memory](#).

### 49.1 Overview

[Table 49.1](#) lists the specifications of the flash memory, and [Figure 49.1](#) is block diagrams of the flash memory related modules.

The I/O pins used in boot mode, see [Table 49.37](#).

The FCU (flash control unit) controls programming and erasure of the flash memory. The FACL (flash application command interface) controls the FCU according to the specified FACL commands.

Regarding the configuration of the code flash memory, see [Figure 49.2](#) to [Figure 49.3](#), and for the configuration of the data flash memory, see [Figure 49.4](#).

**Table 49.1 Specifications of flash memory (1 of 2)**

Item	Code flash memory	Data flash memory
Memory capacity	User area: 1 Mbytes max	Data area: 12 Kbytes
Value after erasure	0xFF	Undefined
Programming/erasing method	<ul style="list-style-type: none"> <li>Programming and erasing the code flash memory and data flash memory, and programming the option-setting memory are handled by the FACL commands specified in the FACL command issuing area (Secure: 0x4010_0000, Non-secure: 0x5010_0000) (self-programming).</li> <li>Programming/erasure through transfer by a serial-programmer via a serial interface (serial programming)</li> </ul>	
Protection	Protects against erroneous rewriting of the flash memory	
Dual bank function	The dual-bank structure makes a safe update possible in cases where programming is suspended. <ul style="list-style-type: none"> <li>Linear mode: the code flash memory is used as one area.</li> <li>Dual mode: the code flash memory is divided into two areas.</li> </ul>	Not available
Block swap function	The block swap structure makes a safe update for a part of Non-secure application possible in case where programming is suspended.	Not available
Background operations (BGOs)	<ul style="list-style-type: none"> <li>The code flash memory can be read while the code flash memory is being programmed or erased.</li> <li>The data flash memory can be read while the code flash memory is being programmed or erased.</li> <li>The code flash memory can be read while the data flash memory is being programmed or erased.</li> </ul>	
Units of programming and erasure	<ul style="list-style-type: none"> <li>Units of programming for the user area: 128 bytes</li> <li>Units of erasure for the user area: Block units</li> </ul>	<ul style="list-style-type: none"> <li>Unit of programming for the data area: 4/8/16 bytes</li> <li>Unit of erasure for the data area: 64/128/256 bytes</li> </ul>
Other functions	Interrupts can be accepted during self-programming. In the initial settings of this MCU, an expansion area of the option-setting memory can be set.	
On-board programming (four types)	Programming/erasure in boot mode (for the SCI interface) <ul style="list-style-type: none"> <li>The asynchronous serial interface (SCI9) is used.</li> <li>The transfer rate is adjusted automatically.</li> </ul> Programming/erasure in boot mode (for the USB interface) <ul style="list-style-type: none"> <li>USBFS is used.</li> <li>Dedicated hardware is not required, so direct connection to a PC is possible.</li> </ul> Programming/erasure in On-chip debug mode <ul style="list-style-type: none"> <li>JTAG or SWD interface is used</li> </ul> Programming and erasure by self-programming <ul style="list-style-type: none"> <li>This allows code flash memory programming/erasure without resetting the system.</li> </ul>	

**Table 49.1 Specifications of flash memory (2 of 2)**

Item	Code flash memory	Data flash memory
Unique ID	A 16-byte ID code provided for each MCU	
FACI command	Program : 128 bytes Block erase: 1 block (8 KB or 32 KB) P/E suspend P/E resume Forced Stop Status Clear Configuration set (16 bytes)	Program: 4/8/16 bytes Block Erase: 1 block (64 bytes) Multi Block Erase: 64/128/256 bytes P/E suspend P/E resume Forced Stop Blank Check: 4 bytes to data flash memory capacity Status Clear Configuration set (4/16 Bytes) Increment Counter: 1bit Refresh Counter Read Counter: 8 Bytes
Security function	Protects against illicit tampering with or reading out of data in flash memory Startup area select setting protection <ul style="list-style-type: none"> <li>• BTFLG and FSUACR registers are protected by the FSPR bit.</li> </ul> Permanent block protect setting protection <ul style="list-style-type: none"> <li>• Code flash memory is permanently protected from programming/erasure operation by the permanent block protect function.</li> </ul> Flash memory protection for TrustZone <ul style="list-style-type: none"> <li>• Protection for flash memory area (P/E)</li> <li>• Protection for flash memory area (read)</li> <li>• Protection for register</li> <li>• Protection during FACI command operation.</li> <li>• Code flash P/E mode entry protection</li> <li>• Data flash configuration area protection</li> <li>• Anti-rollback Counter</li> </ul>	
Safety function	Software protection <ul style="list-style-type: none"> <li>• FACI command protection by FENTRYR register.</li> <li>• Flash memory is protected by FWEPROR register</li> <li>• The user area is protected by the block protect setting</li> </ul> Error protection <ul style="list-style-type: none"> <li>• Error is detected when unintended commands or prohibited settings occur. The FACI command is not accepted after an error detection.</li> </ul> Boot area protection <ul style="list-style-type: none"> <li>• The start-up area select function allows customer to safely update the boot firmware. The size of the start-up area is 8 KB.</li> </ul>	
Interrupt request	<ul style="list-style-type: none"> <li>• FRDYI (flash sequencer ready (processing end)) : Enabled by FRDYIE bit.</li> <li>• FIFERR (flash sequencer error) : Enabled by CFAEIE/CMDLKIE/DFAEIE bits</li> </ul>	
Address conversion	Start-up area select function is supported in linear mode Dual mode and Linear mode <ul style="list-style-type: none"> <li>• Bank swap function is supported in dual mode</li> <li>• Block swap function is supported in linear mode</li> </ul>	

Figure 49.1 shows how modules related to flash memory can be configured. The flash sequencer is configured with the FCU and FACI. The FCU executes basic control for rewriting of the flash memory. The FACI receives FACI commands using peripheral bus, and controls FCU operations accordingly.

In response to a reset, the FACI transfers data from the flash memory to the option byte storage registers.

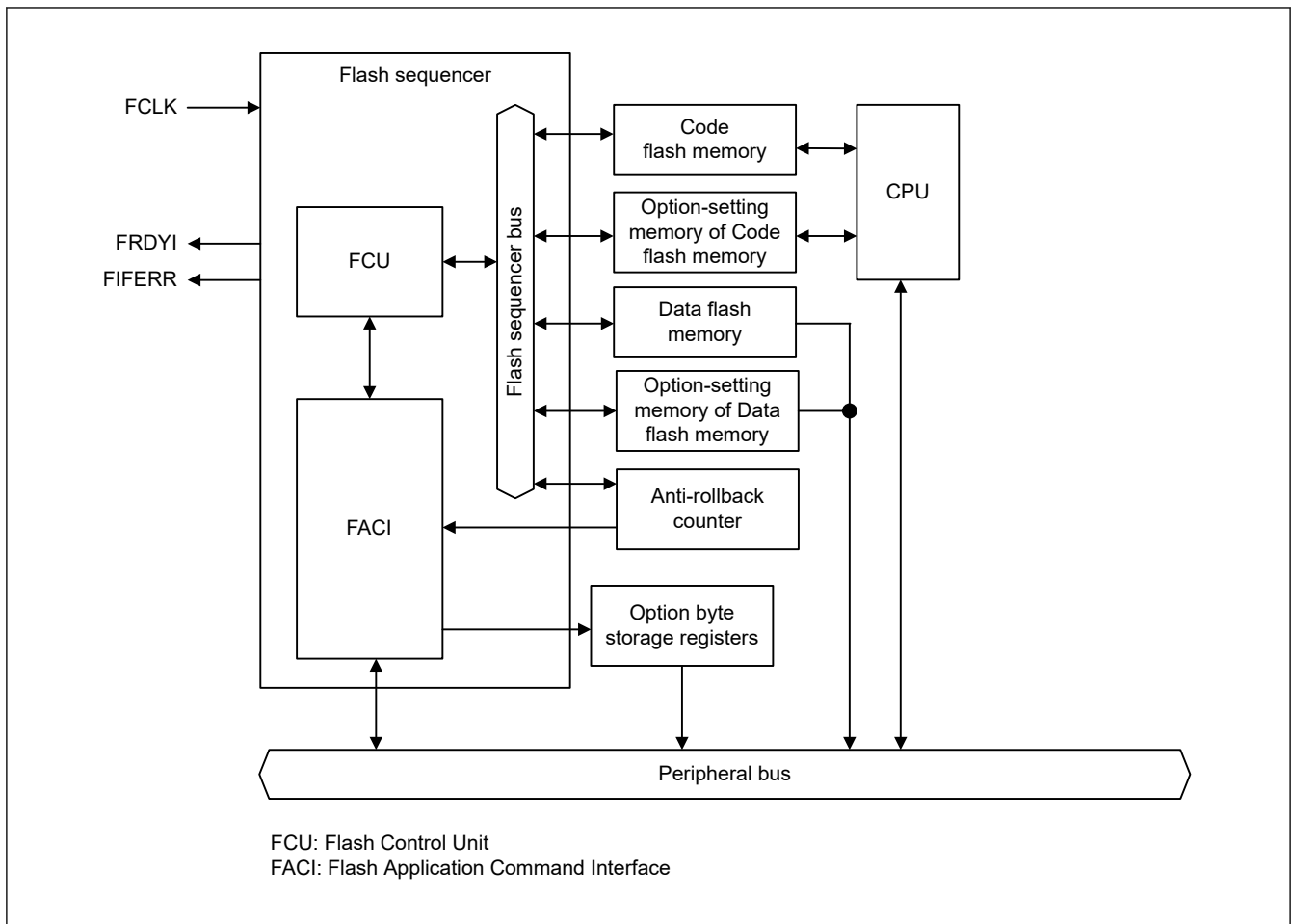


Figure 49.1 Block diagram of flash memory-related modules

## 49.2 Structure of Memory

Figure 49.2 shows the memory map of code flash memory in linear mode. Figure 49.3 shows the memory map of code flash memory in dual mode. This MCU can use the code flash memory as 2 bank areas by using the dual bank function. This dual-bank structure allows a safe update of a program while a user program is running.

The user area of the code flash memory in this MCU is divided into 8 KB and 32 KB blocks, which serve as the units of erasure.

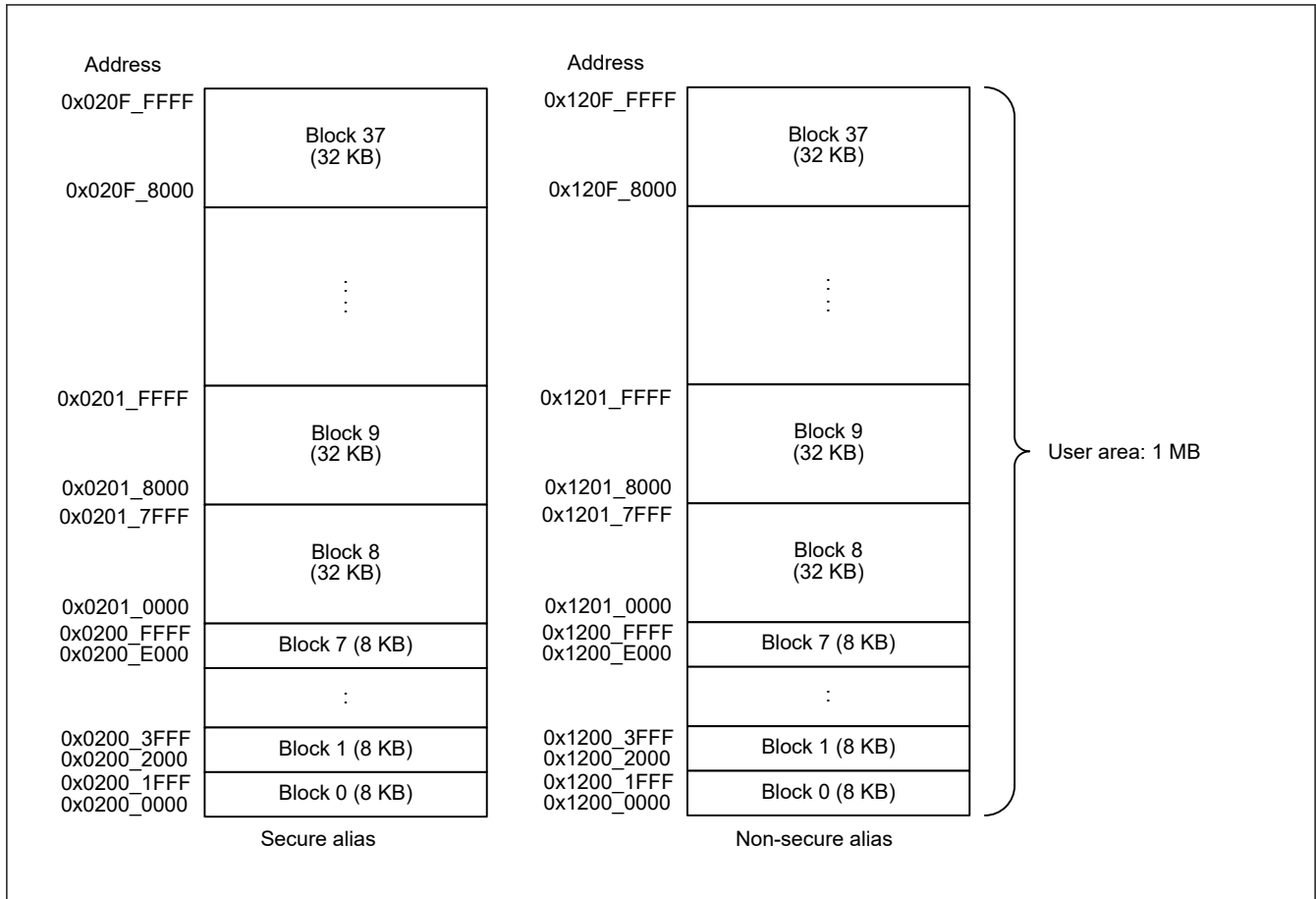
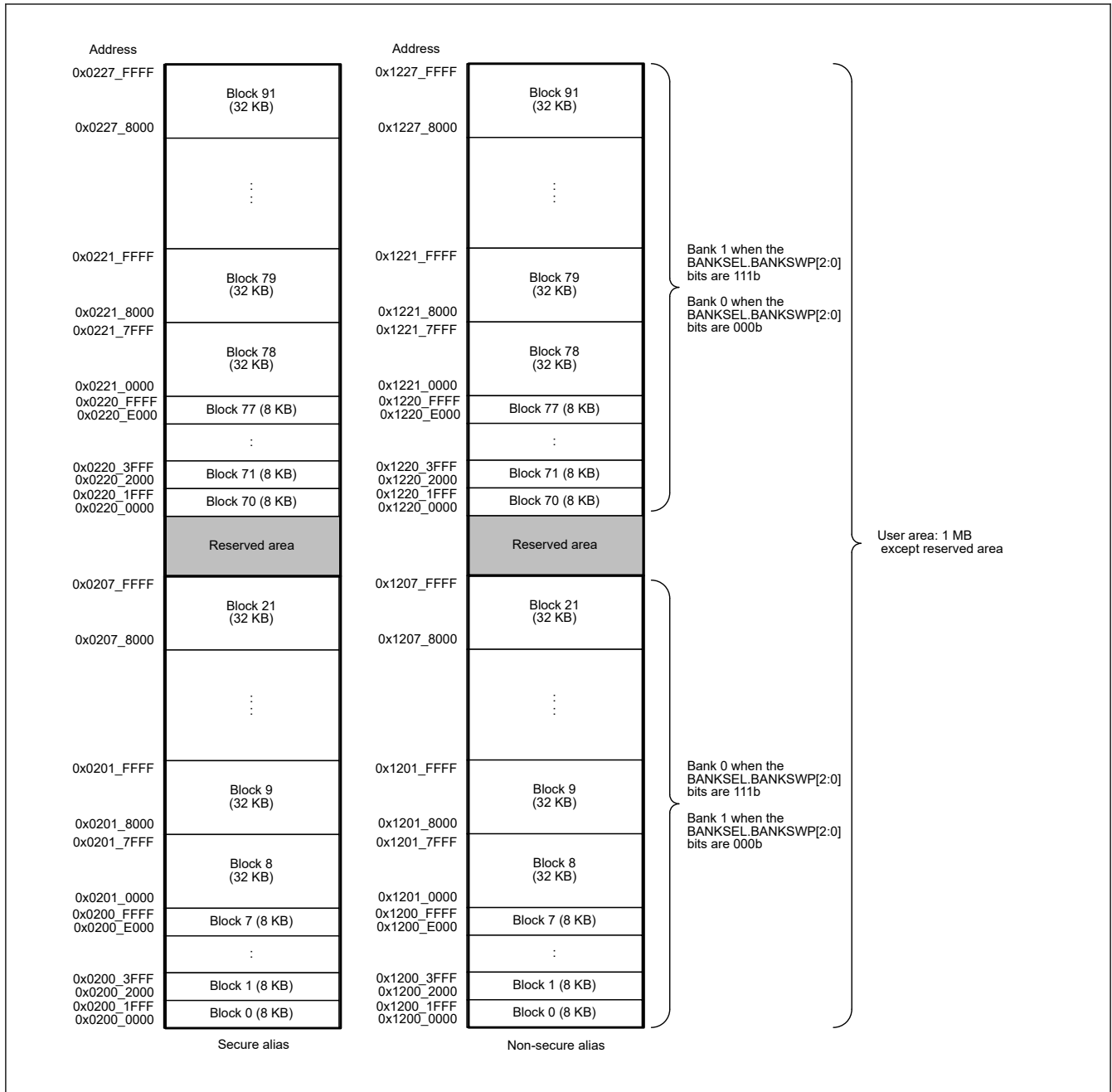


Figure 49.2 Map of the Code Flash Memory in Linear Mode

Table 49.2 Read and programming/erasure address for the code flash memory in Linear Mode

Product	Address	Number of blocks
1 MB product	Secure alias : 0x0200_0000 to 0x020F_FFFF Non-secure alias : 0x1200_0000 to 0x120F_FFFF	0 to 37





**Figure 49.3** Map of the Code Flash Memory in Dual Mode

**Table 49.3** Read and programming/erasure address for the code flash memory in Dual Mode

Product	Address	Number of blocks
1 MB product, lower side bank	Secure alias : 0x0200_0000 to 0x0207_FFFF Non-secure alias : 0x1200_0000 to 0x1207_FFFF	0 to 21
1 MB product, upper side bank	Secure alias : 0x0220_0000 to 0x0227_FFFF Non-secure alias : 0x1220_0000 to 0x1227_FFFF	70 to 91

The data area of the data flash memory in this MCU is divided into 64 byte blocks, with each being a unit for erasure. [Figure 49.4](#) shows the mapping of the data flash memory.

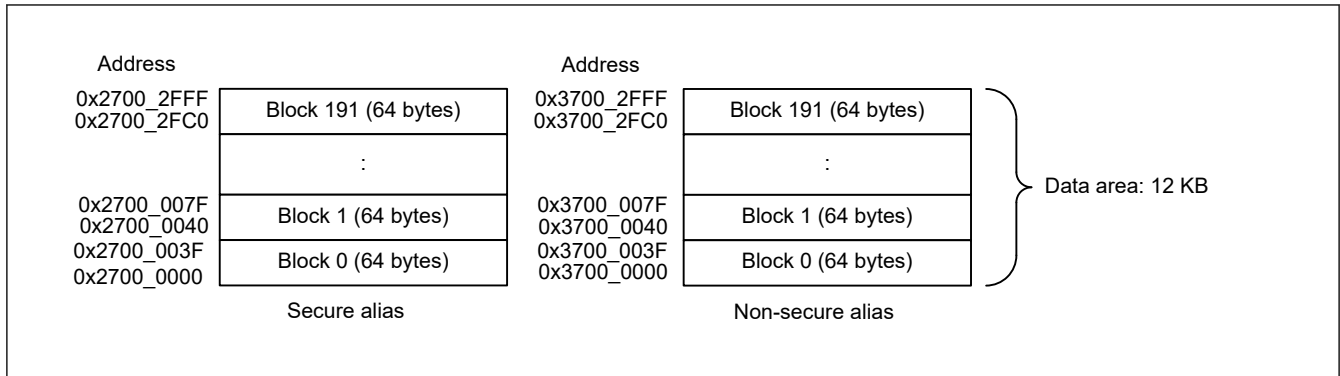


Figure 49.4 Map of the Data Flash Memory

### 49.3 Address Space

Using the hardware interface with flash memory requires access to all registers of the hardware, which is for issuing FACI commands. Table 49.4 provides information about the hardware interface.

Table 49.4 Information on the hardware interface area

Area	Address	Capacity
Area containing various registers of the hardware	See section 49.4. Register Descriptions.	See section 49.4. Register Descriptions.
FACI command-issuing area	Secure alias : 0x4010_0000 Non-secure alias : 0x5010_0000	4 bytes

For the address information of the flash memory, see Figure 49.2.

### 49.4 Register Descriptions

#### 49.4.1 FCACHEE : Flash Cache Enable Register

Base address: FCACHE = 0x4001\_C100  
FCACHE\_NS = 0x5001\_C100

Offset address: 0x000

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FCACHEEN
------------	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----------

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	FCACHEEN	Flash Cache Enable 0: FCACHE is disabled 1: FCACHE is enabled	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See section 49.4.4. FSAR : Flash Security Attribution Register.

#### FCACHEEN bit (Flash Cache Enable)

FCACHEE.FCACHEEN bit enable and disables the function of Flash Cache of FCACHE1, FCACHE2 and FLPF.

FCACHEE.FCACHEEN bit dose not influence for FCACHEIV.FCACHEIV.

When FCACHE is enabled, it works for accesses marked as cacheable.

It is prohibited to disable FCACHE after enabling.

### 49.4.2 FCACHEIV : Flash Cache Invalidate Register

Base address: FCACHE = 0x4001\_C100  
FCACHE\_NS = 0x5001\_C100

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FCACHEIV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FCACHEIV	Flash Cache Invalidate 0: Read: Do not invalidate. Write: The setting is ignored. 1: Invalidate FCACHE is invalidated.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-5, P-TYPE-2

This register is not controlled by any security attribute register.

#### FCACHEIV bit (Flash Cache Invalidate)

When 1 is written to FCACHEIV.FCACHEIV bit, the Flash cache data of FCACHE1, FCACHE2 and FLPF is invalidated.

Invalidate FCACHE with keeping FCACHE enabled after programming or erasing the code flash or the option setting memory.

### 49.4.3 FLWT : Flash Wait Cycle Register

Base address: FCACHE = 0x4001\_C100  
FCACHE\_NS = 0x5001\_C100

Offset address: 0x01C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	FLWT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	FLWT[2:0]	Flash Wait Cycle 0 0 0: 0 wait (ICLK ≤ 48 MHz) 0 0 1: 1 wait (48 MHz < ICLK ≤ 96 MHz) 0 1 0: 2 wait (96 MHz < ICLK ≤ 144 MHz) 0 1 1: 3 wait (144 MHz < ICLK ≤ 192 MHz) 1 0 0: 4 wait (192 MHz < ICLK ≤ 240 MHz) Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#).

#### FLWT[2:0] bits (Flash Wait Cycle)

The Flash Wait Cycle Register (FLWT) sets the access wait count for the flash memory.

For faster clock frequencies, set FLWT.FLWT before changing the clock frequency. For slower clock frequencies, set FLWT.FLWT after changing the clock frequency.

For information on the frequency setting, see [section 8, Clock Generation Circuit](#).

#### 49.4.4 FSAR : Flash Security Attribution Register

Base address: FCACHE = 0x4001\_C100  
FCACHE\_NS = 0x5001\_C100

Offset address: 0x040

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	FACIC MRSA	FACIC MISA	FCKM HZSA	—	—	—	—	—	—	FCAC HESA	FLWT SA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLWTSA	FLWT Security Attribution Target register : FLWT 0: Secure 1: Non-secure	R/W
1	FCACHESA	FCHACHEEN Security Attribution Target register : FCHACHEEN 0: Secure 1: Non-secure	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	FCKMHZSA	FCKMHZ Security Attribution Target register : FCKMHZ 0: Secure 1: Non-secure	R/W
9	FACICMISA	FACI command Issuing Security Attribution Target area : Flash access area 0: Secure 1: Non-secure	R/W
10	FACICMRSA	FACI command Registers Security Attribution Target registers : FASTAT, FAEINT, FRDYIE, FSADDR, FEADDR, FBPROT1, FSTATR, FENTRYR, FSUINTR, FCMDR, FBCCNT, FBCSTAT, FPSADDR, FSUASMON, FCPSR, FPCKAR, FWEPROR 0: Secure 1: Non-secure	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-1, P-TYPE-1

Note: Set the PRCR.PRC4 bit to 1 (write enabled) before rewriting this register.

##### FLWTSA bit (FLWT Security Attribution)

This bit sets the security attribute of FLWT.

##### FCACHESA bit (FCHACHEEN Security Attribution)

This bit sets the security attribute of FCHACHEEN.

##### FCKMHZSA bit (FCKMHZ Security Attribution)

This bit sets the security attribute of Flash access area.

##### FACICMISA bit (FACI command Issuing Security Attribution)

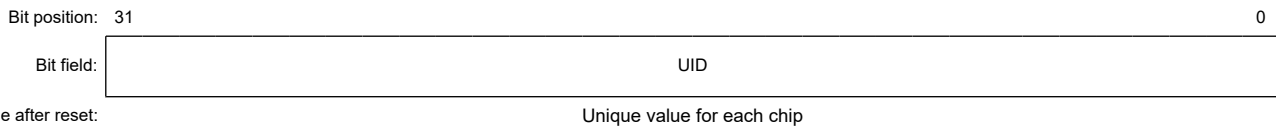
This bit sets the security attribute of Flash access area.

##### FACICMRSA bit (FACI command Registers Security Attribution)

This bit sets the security attribute of FACI command security registers.

### 49.4.5 UIDRn : Unique ID Registers n (n = 0 to 3)

Address: 0x0300\_8190 + n × 4 (Secure)  
0x1300\_8190 + n × 4 (Non-secure)



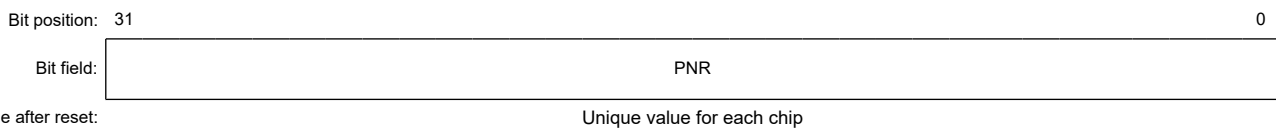
Bit	Symbol	Function	R/W
31:0	UID	Unique ID	R

Note: S-TYPE-5, P-TYPE-5

The UIDRn is a read-only register that stores a 16-byte ID code (unique ID) for identifying the individual MCU. The UIDRn register should be read in 32-bit units. When reading by the signature request command of the serial programming interface, the data is read in order from the data with the large address. That is, the data in 0x0300\_819F is read first, and in 0x0300\_8190 is read last.

### 49.4.6 PNRn : Part Numbering Register n (n = 0 to 3)

Address: 0x0300\_80F0 + n × 4 (Secure)  
0x1300\_80F0 + n × 4 (Non-secure)



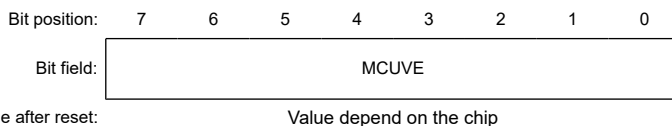
Bit	Symbol	Function	R/W
31:0	PNR	Part Number	R

Note: S-TYPE-5, P-TYPE-5

The PNRn is a read-only register that stores a 16-byte part numbering. The PNRn register should be read in 32-bit units. Each byte corresponds to the ASCII code representation of the product part number as detailed in [Table 1.13](#). The first character ("R", 0x52 in ASCII code) of the part number is stored in the byte with the smallest address (0x0300\_80F0). When reading by the signature request command of the serial programming interface, the data is read in order from the data with the small address. That is, the data in 0x0300\_80F0 is read first, and in 0x0300\_80FF is read last.

### 49.4.7 MCOVER : MCU Version Register

Address: 0x0300\_81B0 (Secure)  
0x1300\_81B0 (Non-secure)



Bit	Symbol	Function	R/W
7:0	MCOVER	MCU Version	R

Note: S-TYPE-5, P-TYPE-5

The MCOVER is a read-only register that stores a MCU version. The MCOVER register should be read in 8-bit units.

### 49.4.8 FWEPROR : Flash P/E Protect Register

Base address: SYSC = 0x4001\_E000  
SYSC\_NS = 0x5001\_E000

Offset address: 0xA54

Bit position: 7 6 5 4 3 2 1 0

Bit field:	—	—	—	—	—	—	FLWE[1:0]
------------	---	---	---	---	---	---	-----------

Value after reset: 0 0 0 0 0 0 1 0

Bit	Symbol	Function	R/W
1:0	FLWE[1:0]	Flash Programming and Erasure 0 0: Prohibits Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing. 0 1: Permits Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing. 1 0: Prohibits Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing. 1 1: Prohibits Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

It is possible that Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command processing are prohibited by software.

The FWEPROR register is initialized by a reset from the following:

- All reset source except for VBATT\_POR reset
- Transition to Deep Software Standby mode
- Transition to Software Standby mode.

#### FLWE[1:0] bits (Flash Programming and Erasure)

The FLWE[1:0] bits are used to set the flash P/E protection. The value after reset is 10b.

If these bits are set to other than 01b that does not allow programming and erasure of the flash memory, the following commands cannot be executed. Issuing any of the following commands leads to setting of the FLWEERR bit in the FSTATR register to 1.

Program / Block Erase / Multi Block Erase / Blank Check / Configuration set command

### 49.4.9 FASTAT : Flash Access Status Register

Base address: FACL = 0x4011\_E000  
FACL\_NS = 0x5011\_E000

Offset address: 0x10

Bit position: 7 6 5 4 3 2 1 0

Bit field:	CFAE	—	—	CMDL K	DFAE	—	—	—
------------	------	---	---	-----------	------	---	---	---

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	DFAE	Data Flash Memory Access Violation Flag 0: No data flash memory access violation has occurred 1: A data flash memory access violation has occurred.	R/W <sup>1</sup>

Bit	Symbol	Function	R/W
4	CMDLK	Command Lock Flag 0: The flash sequencer is not in the command-locked state 1: The flash sequencer is in the command-locked state.	R
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	CFAE	Code Flash Memory Access Violation Flag 0: No code flash memory access violation has occurred 1: A code flash memory access violation has occurred.	R/W <sup>1</sup>

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

Note 1. Only 0 can be written to clear the flag after 1 is read.

The FASTAT register indicates whether a code flash or data flash memory access violation has occurred. If any of the CFAE, CMDLK, and DFAE bits is set to 1, the flash sequencer enters the command-locked state (see [section 49.11.2. Error Protection](#)). To release it from the command-locked state, issue a status clear command or Forced Stop command to the flash sequencer.

### DFAE bit (Data Flash Memory Access Violation Flag)

The DFAE bit indicates whether a data flash memory access violation occurred. When this bit is set to 1, the ILGLERR bit in the FSTATR register is set to 1, placing the flash sequencer in the command-locked state.

[Setting conditions]

FACI commands issued in the data flash P/E mode are as follows:

- The setting of the FSADDR or FEADDR register is the reserved portion of the data area
- The Configuration set command is issued while the setting of the FSADDR register is out of 0x2703\_0050 to 0x2703\_03FF.
- FACI command is issued to data flash memory with the wrong security attribution.

[Clearing conditions]

- When this bit is written as 0 after it is set to 1
- When the flash sequencer starts to process the Status Clear or Forced Stop command.

### CMDLK bit (Command Lock Flag)

The CMDLK bit indicates that the flash sequencer is in the command-locked state.

[Setting conditions]

- The flash sequencer detects an error and enters the command-locked state.

[Clearing conditions]

- When the flash sequencer starts to process the Status Clear or Forced Stop command.

### CFAE bit (Code Flash Memory Access Violation Flag)

The CFAE bit indicates whether a code flash memory access violation has occurred. When this bit is set to 1, the ILGLERR bit in the FSTATR register is set to 1, placing the flash sequencer in the command-locked state.

[Setting conditions]

FACI commands issued in the code flash P/E mode are as follows:

- The setting of the FSADDR register is the reserved portion of the user area
- The Configuration set command is issued while the setting of the FSADDR register is from 0x0200\_A100 to 0x0200\_A2F0 in self-programming mode
- FACI command is issued to code flash memory with the wrong security attribution.

[Clearing conditions]

- When this bit is written as 0 after it is set to 1

- When the flash sequencer starts to process the Status Clear or Forced Stop command.

#### 49.4.10 FAEINT : Flash Access Error Interrupt Enable Register

Base address: FACL = 0x4011\_E000  
FACL\_NS = 0x5011\_E000

Offset address: 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CFAEIE	—	—	CMDLKIE	DFAEIE	—	—	—

Value after reset: 1 0 0 1 1 0 0 0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	DFAEIE	Data Flash Memory Access Violation Interrupt Enable 0: Generation of an FIFERR interrupt request is disabled when FASTAT.DFAE is set to 1 1: Generation of an FIFERR interrupt request is enabled when FASTAT.DFAE is set to 1.	R/W
4	CMDLKIE	Command Lock Interrupt Enable 0: Generation of an FIFERR interrupt request is disabled when FASTAT.CMDLK is set to 1 1: Generation of an FIFERR interrupt request is enabled when FASTAT.CMDLK is set to 1.	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	CFAEIE	Code Flash Memory Access Violation Interrupt Enable 0: Generation of an FIFERR interrupt request is disabled when FASTAT.CFAE is set to 1 1: Generation of an FIFERR interrupt request is enabled when FASTAT.CFAE is set to 1.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#).

The FAEINT register enables or disables generation of a flash access error (FIFERR) interrupt request.

##### DFAEIE bit (Data Flash Memory Access Violation Interrupt Enable)

The DFAEIE bit enables or disables generation of an FIFERR interrupt request when a data flash memory access violation occurs, setting the DFAE bit in the FASTAT register to 1.

##### CMDLKIE bit (Command Lock Interrupt Enable)

The CMDLKIE bit enables or disables generation of an FIFERR interrupt request when the flash sequencer enters the command-locked state, setting the CMDLK bit in the FASTAT register to 1.

##### CFAEIE bit (Code Flash Memory Access Violation Interrupt Enable)

The CFAEIE bit enables or disables generation of an FIFERR interrupt request when a code flash memory access violation occurs, setting the CFAE bit in the FASTAT register to 1.

#### 49.4.11 FRDYIE : Flash Ready Interrupt Enable Register

Base address: FACL = 0x4011\_E000  
FACL\_NS = 0x5011\_E000

Offset address: 0x18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	FRDYIE

Value after reset: 0 0 0 0 0 0 0 0 0



Bit	Symbol	Function	R/W
0	FRDYIE	Flash Ready Interrupt Enable 0: Generation of an FRDY interrupt request is disabled 1: Generation of an FRDY interrupt request is enabled.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

The FRDYIE register enables or disables generation of a flash ready (FRDY) interrupt request.

### FRDYIE bit (Flash Ready Interrupt Enable)

The FRDYIE bit enables or disables generation of an FRDY interrupt request when the FRDY bit in the FSTATR register is changed from 0 to 1 on completion of processing by the flash sequencer of the Program, Block Erase, Multi Block Erase, Blank Check, and Configuration set command.

## 49.4.12 FSADDR : FACI Command Start Address Register

Base address: FACI = 0x4011\_E000  
FACI\_NS = 0x5011\_E000

Offset address: 0x30

Bit position: 31 0

Bit field: FSADDR[31:0]

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	FSADDR[31:0]	Start Address for FACI Command Processing	R/W*1

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

Note 1. These bits can be written when the FRDY bit in the FSTATR register is 1. Writing to these bits are ignored when the FRDY bit is 0.  
Note that b0 and b1 are read-only.

**Table 49.5 FACI command address boundary**

Command	Address Boundary
Program (code flash memory)	128 byte
Program (data flash memory)	4, 8, 16 byte
Block Erase (code flash memory)	8 KB or 32 KB
Block Erase (data flash memory)	64 byte
Multi Block Erase (data flash memory)	64 byte
Blank Check (data flash memory)	4 byte
For code flash configuration set	16 byte
For data flash configuration set	4, 16 byte

The FSADDR register specifies the address where the target area for command processing starts when the FACI command for Program, Block Erase, Multi Block Erase, Blank Check, or Configuration set is issued.

The FSADDR value is initialized when the SUNIT bit in the FSUNITR register is set to 1. It is also initialized by a reset.

### FSADDR[31:0] bits (Start Address for FACI Command Processing)

The FSADDR[31:0] bits specify the start address for FACI command processing. Bits [31:29] and [27:24] are ignored in FACI command processing for the code flash memory. Bits [31:29] and [27:13] are ignored in FACI command processing for the data flash memory. Bit [28] is used the security alias for flash sequencer command. When bit [28] is "0", flash sequencer can update secure region with secure access. When bit [28] is "1", flash sequencer can update non-security region with secure and non-secure access. Bits associated with the address bits of lower order than the address boundary listed in [Table 49.5](#) are also ignored.

For information on the addresses of the code flash memory and the data flash memory, see [section 49.2. Structure of Memory](#).

For information on the addresses of the configuration setting, see [section 49.9.3.15. Configuration Set Command](#).

### 49.4.13 FEADDR : FACL Command End Address Register

Base address: FACL = 0x4011\_E000  
FACL\_NS = 0x5011\_E000

Offset address: 0x34



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	FEADDR[31:0]	End Address for FACL Command Processing	R/W <sup>1</sup>

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

Note 1. These bits can be written when the FRDY bit in the FSTATR register is 1. Writing to these bits are ignored when the FRDY bit is 0. Note that bit [0] and bit [1] are read-only.

The FEADDR register specifies the end address of the target area for Multi Block Erase and Blank Check command processing. When incremental mode is selected as the addressing mode for Blank Checking (when FBCCNT.BCDIR = 0), the address specified in the FSADDR register should be equal to or smaller than the address in the FEADDR register. Conversely, the address in the FSADDR register should be equal to or larger than the address in the FEADDR register when decremental mode is selected as the addressing mode for Blank Check (i.e. when FBCCNT.BCDIR = 1). If the BCDIR, FSADDR, and FEADDR bit settings are inconsistent with the specified rules, the flash sequencer enters the command-locked state (see [section 49.11.2. Error Protection](#)).

The FEADDR value is initialized when the SUINIT bit in the FSUINITR register is set to 1. It is also initialized by a reset.

#### FEADDR[31:0] bits (End Address for FACL Command Processing)

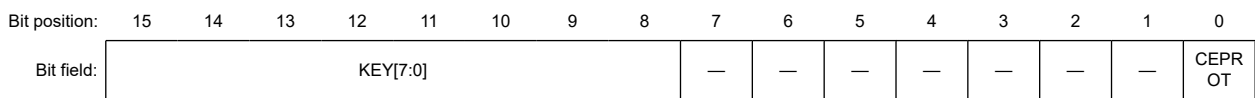
The FEADDR[31:0] bits specify the end address for Multi Block Erase and Blank Check command processing. In command processing, bits 31 to 17 and any bits that do not reach the address boundaries listed in the [section 49.4.12. FSADDR : FACL Command Start Address Register](#) are ignored.

For information on the addresses of the flash memory, see [section 49.2. Structure of Memory](#).

### 49.4.14 FMEPROT : Flash P/E Mode Entry Protection Register

Base address: FACL = 0x4011\_E000

Offset address: 0x44



Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 x 1

Bit	Symbol	Function	R/W
0	CEPROT	Code Flash P/E Mode Entry Protection 0: FENTRYC bit is not protected 1: FENTRYC bit is protected.	R/W <sup>1</sup> *2 *4
1	—	The read value is undefined. The write value should be 0.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	W <sup>3</sup>

Note: S-TYPE-6, P-TYPE-2

Note 1. Writing to this bit is only possible when the FRDY bit in the FSTATR register is 1. Writing to this bit while the FRDY bit = 0 is ignored.

Note 2. Writing to this bit is only possible when 16 bits are written and the value written to the KEY bits is 0xD9.

Note 3. Written values are not retained by these bits (always read as 0x00).

Note 4. Only Secure access can write to this register. Both Secure access and Non-secure read access are allowed. Non-secure write access is denied, but TrustZone access error is not generated.

### CEPROT bit (Code Flash P/E Mode Entry Protection)

The CEPROT bit specifies the protection setting of the FRNTRYC bit in the FENTRYR register.

[Setting condition]

- 1 being written to the CEPROT bit while writing to FMEPROT is enabled.

[Clearing condition]

- 0 being written to the CEPROT bit while writing to FMEPROT is enabled.

### 49.4.15 FCNTSELR : Flash Counter Select Register

Base address: FACL = 0x4011\_E000

Offset address: 0x048

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CNTSEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CNTSEL[2:0]	Counter Select 0 0 0: Anti-rollback counter is not selected 0 0 1: ARC_SEC is selected. Counter size is 64 bits. 0 1 1: Anti-rollback counter is not selected. 1 x x: ARC_NSEC is selected. Counter configuration is 64 or 256 bits. See , <a href="#">Table 49.6</a> <a href="#">Table 49.7</a> and <a href="#">Table 49.8</a> in details counter select condition. Others: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-6, P-TYPE-2

Note: If the security attribution is configured as Secure:

- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:

- Secure and Non-secure access are allowed.

### CNTSEL[2:0] bits (Counter Select)

FCNTSELR specifies 2 types of anti-rollback counter (ARC\_SEC or ARC\_NSEC) in "Increment Counter", "Refresh Counter", or "Read Counter" command.

FCNTSELR value is initialized when SUNIT bit in FSUINTR is set to "1"

Note: This bit can be written when the FRDY bit in the FSTATR register is 1. Writing to this bit is ignored when the FRDY bit is 0.

Before using the ARC\_NSEC counter, it is necessary to set whether the counter is to be used with 4line x 64 bits or 1line x 256 bits using the Data Flash configuration set command. The bits to be set are CNF\_ARCNS0 and CNF\_ARCNS1. If the CNF\_ARCNS1 bit is set to 1, the ARC\_NSEC counter cannot be used. Once the counter type is decided, it cannot be changed.

**Table 49.6 Select condition of ARC\_NSEC (64 bits) in Increment/Refresh/Read Counter Command**

FCNTSELR[2:0]	Target Counter	Counter Configuration
100	ARC_NSEC[63:0]	Counter number is 4. Counter size is 64bit.
101	ARC_NSEC[127:64]	
110	ARC_NSEC[191:128]	
111	ARC_NSEC[255:192]	

**Table 49.7 Select condition of ARC\_NSEC (256 bits) in Increment/Refresh Counter Command**

FCNTSELR[2:0]	Target Counter	Counter Configuration
100	ARC_NSEC[255:0]	Counter number is 1. Counter size is 256bit.
101		
110		
111		

**Table 49.8 Select condition of ARC\_NSEC (256 bits) in Read Counter Command**

FCNTSELR[2:0]	Target Counter	Counter Configuration
100	ARC_NSEC[63:0]	Counter number is 1. Counter size is 256bit.
101	ARC_NSEC[127:64]	
110	ARC_NSEC[191:128]	
111	ARC_NSEC[255:192]	

#### 49.4.16 FCNTDATARn : Flash Counter Data Register n (n = 0, 1)

Base address: FACI = 0x4011\_E000

Offset address: 0x04C + 0x004 × n (n = 0, 1)

Bit position: 31

0

Bit field:



Value after reset:

Unique value for each chip

Bit	Symbol	Function	R/W
31:0	CNTRDAT	Counter Read Data Reading data of 32 bits (LSB side of counter values) is output to FCNTDATAR0. Reading data of 32 bits (MSB side of counter values) is output to FCNTDATAR1.	R

Note: S-TYPE-6, P-TYPE-2

FCNTDATAR0 and FCNTDATAR1 indicate anti-rollback counter values (64bits) that read by "Read Counter" command.

#### CNTRDAT bit (Counter Read Data)

[Setting condition]

- FACI has accepted "Read Counter" command.

[Clearing condition]

- FACI has accepted flash sequencer command except for "Read Counter" command.
- Flash sequencer enters "Command Lock" state

### 49.4.17 FBPROT0 : Flash Block Protection Register

Base address: FACL = 0x4011\_E000  
FACL\_NS = 0x5011\_E000

Offset address: 0x78

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	BPCN 0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BPCN0	Block Protection for Non-secure Cancel 0: Block protection is enabled 1: Block protection is disabled.	R/W <sup>*1 *2</sup>
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	W <sup>*3</sup>

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

Note 1. This bit can be written when the FRDY bit in the FSTATR register is 1. Writing to this bit is ignored when the FRDY bit is 0.

Note 2. Writing to this bit is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0x78.

Note 3. Written values are not retained by these bits (always read as 0x00).

The FBPROT0 register is used to disable the block protect function for Non-secure. When the block protect setting is locked by the permanent block setting, it cannot be disabled by this register.

The FBPROT0 value is initialized when the SUNIT bit in the FSUINITR is set to 1, because the FENTRYR value is initialized to 0x0000. It is also initialized by a reset.

#### BPCN0 bit (Block Protection for Non-secure Cancel)

The BPCN0 bit disables the block protect setting for Non-secure function.

[Setting condition]

- When the write-enabling conditions are satisfied and the FENTRYR is not 0x0000, write 1 to this bit.

[Clearing conditions]

- 8 bits being written to FBPROT0 while the FRDY bit is 1.
- A value other than 0x78 specified in the KEY bits and 16 bits are written to FBPROT0 while the FRDY bit is 1.
- 0 being written to the BPCN0 bit while writing to FBPROT0 is enabled.
- The FENTRYR register value is 0x0000.

### 49.4.18 FBPROT1 : Flash Block Protection for Secure Register

Base address: FACL = 0x4011\_E000

Offset address: 0x7C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]								—	—	—	—	—	—	—	BPCN 1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BPCN1	Block Protection for Secure Cancel 0: Block protection is enabled 1: Block protection is disabled.	R/W <sup>*1 *2</sup>
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	KEY[7:0]	Key Code	W <sup>3</sup>

Note: S-TYPE-6, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

Note 1. Writing to this bit is only possible when the FRDY bit in the FSTATR register is 1. Writing to this bit while FRDY bit = 0 is ignored.

Note 2. Writing to this bit is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0xB1.

Note 3. Written values are not retained by these bits (always read as 0x00).

The FBPROT1 register is used to disable the block protect function for Secure developer. When the block protect setting is locked by the permanent block setting, it cannot be disabled by this register.

The FBPROT1 value is initialized when the SUNIT bit in the FSUNITR is set to 1, because the FENTRYR value is initialized to 0x0000. It is also initialized by a reset.

### BPCN1 bit (Block Protection for Secure Cancel)

The BPCN1 bit disables the block protect setting for Secure function.

[Setting condition]

- When the write-enabling conditions are satisfied and the FENTRYR is not 0x0000, write 1 to BPCN1.

[Clearing conditions]

- 8 bits being written to FBPROT1 while the FRDY bit is 1.
- A value other than 0xB1 specified in the KEY bits and 16 bits are written to FBPROT1 while the FRDY bit is 1.
- 0 being written to the BPCN1 bit while writing to FBPROT1 is enabled.
- The FENTRYR register value is 0x0000.

## 49.4.19 FSTATR : Flash Status Register

Base address: FACL = 0x4011\_E000  
FACL\_NS = 0x5011\_E000

Offset address: 0x80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	ILGCO MERR	FESE TERR	SECE RR	OTER R	TZFE RR	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FRDY	ILGLE RR	ERSE RR	PRGE RR	SUSR DY	DBFU LL	ERSS PD	PRGS PD	—	FLWE ERR	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0.	R
6	FLWEERR	Flash Write/Erase Protect Error Flag 0: An error has not occurred 1: An error has occurred.	R
7	—	This bit is read as 0.	R
8	PRGSPD	Programming Suspend Status Flag 0: The flash sequencer is not in the programming suspension processing state or programming suspended state 1: The flash sequencer is in the programming suspension processing state or programming suspended state.	R
9	ERSSPD	Erase Suspend Status Flag 0: The flash sequencer is not in the erasure suspension processing state or the erasure suspended state 1: The flash sequencer is in the erasure suspension processing state or the erasure suspended state.	R

Bit	Symbol	Function	R/W
10	DBFULL	Data Buffer Full Flag 0: The data buffer is empty 1: The data buffer is full.	R
11	SUSRDY	Suspend Ready Flag 0: The flash sequencer cannot receive P/E suspend commands 1: The flash sequencer can receive P/E suspend commands.	R
12	PRGERR	Programming Error Flag 0: Programming has completed successfully 1: An error has occurred during programming.	R
13	ERSERR	Erase Error Flag 0: Erasure has completed successfully 1: An error has occurred during erasure.	R
14	ILGLERR	Illegal Command Error Flag 0: The flash sequencer has not detected an illegal FACI command or illegal flash memory access 1: The flash sequencer has detected an illegal FACI command or illegal flash memory access.	R
15	FRDY	Flash Ready Flag 0: Program, Block Erase, Multi Block Erase, P/E suspend, P/E resume, Forced Stop, Blank Check, Configuration set, Increment counter, Refresh counter, or Read counter command processing is in progress. 1: None of the above is in progress.	R
18:16	—	These bits are read as 0.	R
19	TZFERR	TrustZone Filter Error 0: A TrustZone filter error has not been detected. 1: A TrustZone filter error has been detected.	R
20	OTERR	Other Error 0: An error has not been detected. 1: An error has been detected.	R
21	SECERR	Security Error 0: A write protection error against MSUASMON.FSPR bit has not been detected. 1: A write protection error against MSUASMON.FSPR bit has been detected.	R
22	FESETERR	FENTRY Setting Error 0: A setting error in the FENTRYR register has not been detected. 1: A setting error in the FENTRYR register has been detected.	R
23	ILGCOMERR	Illegal Command Error 0: An illegal FACI command error has not been detected. 1: An illegal FACI command error has been detected.	R
31:24	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

The FSTATR register indicates the state of the flash sequencer.

### FLWEERR flag (Flash Write/Erase Protect Error Flag)

The FLWEERR flag indicates a violation of the flash memory overwrite protection setting in the FWEPROR register. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The flash sequencer starts processing the Forced Stop command.

### PRGSPD flag (Programming Suspend Status Flag)

The PRGSPD flag indicates that the flash sequencer is in the programming suspension processing state or programming suspended state.

[Setting condition]

- The flash sequencer starts processing in response to the programming suspend command.

[Clearing conditions]

- Reception of the P/E resume command by the flash sequencer (after write access to the FACI command-issuing area is complete)
- The flash sequencer starts processing the Forced Stop command.

### **ERSSPD flag (Erasure Suspend Status Flag)**

The ERSSPD flag indicates that the flash sequencer is in the erasure suspension processing state or erasure suspended state.

[Setting condition]

- The flash sequencer starts processing in response to an erasure suspend command.

[Clearing condition]

- Reception of the P/E resume command by the flash sequencer (after write access to the FACI command-issuing area is complete)
- The flash sequencer starts processing of the Forced Stop command.

### **DBFULL flag (Data Buffer Full Flag)**

The DBFULL flag indicates the state of the data buffer when the program command is issued. The flash sequencer incorporates a buffer for write data (data buffer). When data for writing to the flash memory are written to the FACI command-issuing area while the data buffer is full, the flash sequencer inserts a wait cycle in the peripheral bus.

[Setting condition]

- The data buffer becomes full while program commands are issued.

[Clearing condition]

- The data buffer becomes empty.

### **SUSRDY flag (Suspend Ready Flag)**

The SUSRDY flag indicates whether the flash sequencer can receive a P/E suspend command.

[Setting condition]

- After starting programming/erasure processing, the flash sequencer enters a state in which P/E suspend commands can be received.

[Clearing conditions]

- Reception of the P/E suspend command or Forced Stop command by the flash sequencer (after write access to the FACI command-issuing area is complete)
- During programming or erasure, the flash sequencer enters the command-locked state
- Programming or erasure has completed.

### **PRGERR flag (Programming Error Flag)**

The PRGERR flag indicates the result of programming of the flash memory. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- An error has occurred during programming.

[Clearing condition]

- The flash sequencer starts processing of the Status Clear or Forced Stop command.



### ERSERR flag (Erasure Error Flag)

The ERSERR flag indicates the result of erasure of the flash memory. When this flag is 1, the flash sequencer is in the command-locked state.

[Setting condition]

- An error has occurred during erasure.

[Clearing condition]

- The flash sequencer starts processing of the Status Clear or Forced Stop command.

### ILGLERR flag (Illegal Command Error Flag)

The ILGLERR flag indicates that the flash sequencer has detected an illegal FACI command or flash memory access. If this flag is 1, the flash sequencer is in the command-locked state.

[Setting conditions]

- See [section 49.11.2. Error Protection](#).

[Clearing condition]

- The flash sequencer starts processing of the Status Clear or Forced Stop command.

### FRDY flag (Flash Ready Flag)

The FRDY flag indicates the command processing state of the flash sequencer.

[Setting conditions]

- The flash sequencer completes command processing
- The flash sequencer receives a P/E suspend command and suspends programming of the flash memory
- The flash sequencer received the Forced Stop command and ended command processing.

Note: In the case of program command processing, the FRDY flag might be set to 1 even if the flash sequencer does not complete command processing. For details, see [section 49.9.3.7. Program Command](#).

[Clearing conditions]

- The flash sequencer received an FACI command
- For Program and Configuration setting, the first write access to the FACI command-issuing area
- For other commands, the last write access to the FACI command-issuing area.

### TZFERR flag (TrustZone Filter Error)

This flag indicates the TrustZone protection error in the FACI command. See [Table 49.30](#). When this bit is "1", flash sequencer enters "Command Lock" state.

[Setting conditions]

- An error has been detected.

[Clearing conditions]

- Status Clear or Forced Stop command processing is completed.

### OTERR flag (Other Error)

This flag indicates that an FACI command has been issued when the condition of accepting commands is not satisfied. See [Table 49.30](#). When this flag is 1, the flash sequencer is in the command-lock state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The status clear or forced stop command processing is complete.

**SECERR flag (Security Error)**

This flag indicates that write protection by the MSUASMON.FSPR bit is violated. See [Table 49.30](#). When this flag is 1, the flash sequencer is in the command-lock state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The status clear or forced stop command processing is complete.

**FESETERR flag (FENTRY Setting Error)**

This flag indicates that a value of 0xAA81 is written in the FENTRYR register or the value in the FENTRYR register differs when P/E is suspended and resumed.

See [Table 49.30](#). When this flag is 1, the flash sequencer is in the command-lock state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The status clear or forced stop command processing is complete.

**ILGCOMERR flag (Illegal Command Error)**

This flag indicates that the flash sequencer has detected an illegal FACI command.

See [Table 49.30](#). When this flag is 1, the flash sequencer is in the command-lock state.

[Setting condition]

- An error has occurred.

[Clearing condition]

- The status clear or forced stop command processing is complete.

**49.4.20 FENTRYR : Flash P/E Mode Entry Register**

Base address: FACL = 0x4011\_E000  
 FACL\_NS = 0x5011\_E000

Offset address: 0x84



Bit	Symbol	Function	R/W
0	FENTRYC	Code Flash P/E Mode Entry 0: Code flash is in read mode 1: Code flash is in P/E mode.	R/W <sup>*1,2</sup>
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	FENTRYD	Data Flash P/E Mode Entry 0: Data flash is in read mode 1: Data flash is in P/E mode.	R/W <sup>*1,2</sup>
15:8	KEY[7:0]	Key Code	W <sup>*3</sup>

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

Note 1. These bits can be written when the FRDY bit in the FSTATR register is 1. Writing to these bits are ignored when the FRDY bit is 0.

Note 2. Writing to these bits is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0xAA.

Note 3. Written values are not retained by these bits (always read 0x00).

FENTRYR is used to specify code flash P/E mode or data flash P/E mode. To specify the code flash P/E mode or data flash P/E mode so that the flash sequencer can receive FACI commands, set either the FENTRYD or FENTRYC bit to 1 to place the flash sequencer in P/E mode.

FENTRYR is initialized when the SUNIT bit in FSUINTR is set to 1. It is also initialized by a reset.

Note: Writing a value of 0XAA81 to this register causes the ILGLERR bit in the FSTATR register to be set to 1, resulting in the flash sequencer being placed in the command-locked state.

**FENTRYC bit (Code Flash P/E Mode Entry)**

The FENTRYC bit specifies P/E mode for the code flash memory.

[Setting condition]

- Write 1 to the FENTRYC bit while writing to FENTRYR is enabled and FENTRYR is 0x0000.

[Clearing conditions]

- Write 8 bits to FENTRYR while the FRDY bit is 1
- A value other than 0xAA is specified in the KEY[7:0] bits and 16 bits are written to FENTRYR while the FRDY bit is 1
- Write 0 to the FENTRYC bit while writing to FENTRYR is enabled
- Write to FENTRYR while writing is enabled and its value is other than 0x0000
- The protection of FMEPROT register is enabled.

**FENTRYD bit (Data Flash P/E Mode Entry)**

The FENTRYD bit specifies P/E mode for the data flash memory.

[Setting condition]

- Write 1 to the FENTRYD bit while writing to FENTRYR is enabled and FENTRYR is 0x0000.

[Clearing conditions]

- Write 8 bits to FENTRYR while the FRDY bit is 1
- Writing of 16 bits to FENTRYR with a value other than 0xAA specified for the KEY[7:0] bits while the FRDY bit is 1
- Write 0 to the FENTRYD bit while writing to FENTRYR is enabled
- Write to FENTRYR while writing is enabled and its value is other than 0x0000.

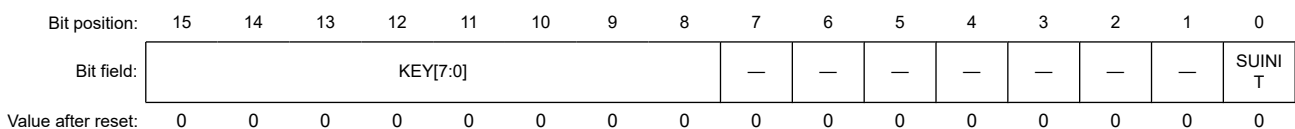
**KEY[7:0] bits (Key Code)**

The KEY[7:0] bits control writing permission to the FENTRYD or FENTRYC bits.

**49.4.21 FSUINTR : Flash Sequencer Setup Initialization Register**

Base address: FACL = 0x4011\_E000  
 FACL\_NS = 0x5011\_E000

Offset address: 0x8C



Bit	Symbol	Function	R/W
0	SUNIT	Set-Up Initialization 0: The FSADDR, FEADDR, FBPROT0, FBPROT1, FENTRYR, FBCCNT, FCPSR, and FCNTSELR flash sequencer setup registers keep their current values 1: The FSADDR, FEADDR, FBPROT0, FBPROT1, FENTRYR, FBCCNT, FCPSR, and FCNTSELR flash sequencer setup registers are initialized.	R/W <sup>*1*2</sup>
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
15:8	KEY[7:0]	Key Code	W*3

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

Note 1. This bit can be written when the FRDY bit in the FSTATR register is 1. Writing to this bit is ignored when the FRDY bit is 0.

Note 2. Writing to these bits is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0x2D.

Note 3. Written values are not retained by these bits (always read 0x00).

FSUINTR is used for initialization of the flash sequencer setup.

### SUINIT bit (Set-Up Initialization)

The SUINIT bit initializes the following flash sequencer setup registers:

- FSADDR
- FEADDR
- FBPROT0
- FBPROT1
- FENTRYR
- FBCCNT
- FCPSR
- FCNTSELR

### KEY[7:0] bits (Key Code)

The KEY[7:0] bits control writing permission to the SUINIT bit.

## 49.4.22 FCMR : FACL Command Register

Base address: FACL = 0x4011\_E000  
FACL\_NS = 0x5011\_E000

Offset address: 0xA0

Bit position: 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

Bit field: 

CMDR[7:0]	PCMDR[7:0]
-----------	------------

Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	PCMDR[7:0]	Pre-command Flag The command just before the latest command is stored.	R
15:8	CMDR[7:0]	Command Flag The latest command is stored.	R

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

FCMR records the two most recent commands accepted by the flash sequencer.

### PCMDR[7:0] bits (Pre-command Flag)

The PCMDR[7:0] bits indicate the command received immediately before the latest command received by the flash sequencer.

### CMDR[7:0] bits (Command Flag)

The CMDR[7:0] bits indicate the latest command received by the flash sequencer.

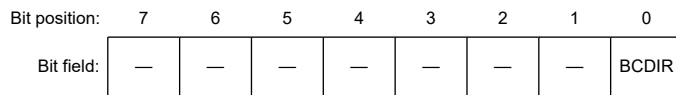
**Table 49.9 States of FCMDR after receiving commands**

Command	CMDR	PCMDR
Program	0xE8	Previous command
Block erase	0xD0	0x20
Multi block erase	0xD0	0x21
P/E suspend	0xB0	Previous command
P/E resume	0xD0	Previous command
Status Clear	0x50	Previous command
Forced Stop	0xB3	Previous command
Blank Check	0xD0	0x71
Configuration set	0x40	Previous command
Increment Counter	0xD0	0x35
Refresh Counter	0xD0	0x37
Read Counter	0xD0	0x39

### 49.4.23 FBCCNT : Blank Check Control Register

Base address: FACL = 0x4011\_E000  
 FACL\_NS = 0x5011\_E000

Offset address: 0xD0



Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	BCDIR	Blank Check Direction 0: Blank checking is executed from the lower addresses to the higher addresses (incremental mode) 1: Blank checking is executed from the higher addresses to the lower addresses (decremental mode).	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

FBCCNT specifies the addressing mode in processing the Blank Check command. FBCCNT is initialized when the SUNIT bit in FSUINTR is set to 1. It is also initialized by a reset.

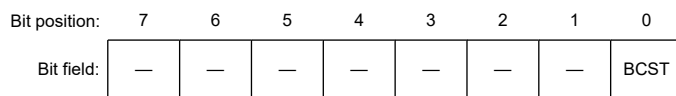
#### BCDIR bit (Blank Check Direction)

The BCDIR bit specifies the addressing mode for Blank Check.

### 49.4.24 FBCSTAT : Blank Check Status Register

Base address: FACL = 0x4011\_E000  
 FACL\_NS = 0x5011\_E000

Offset address: 0xD4



Value after reset: 0 0 0 0 0 0 0 0



Bit	Symbol	Function	R/W
14:0	—	These bits are read as 0. The write value should be 0.	R
15	FSPR	Protection Programming Flag to set Boot Flag and Startup Area Control 0: Protected state 1: Non-protected state.	R
30:16	—	These bits are read as 0. The write value should be 0.	R
31	BTFLG	Flag of Startup Area Select for Boot Swap 0: The startup area is the alternate block (block 1) 1: The startup area is the default block (block 0).	R

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#).

### FSPR bit (Protection Programming Flag to set Boot Flag and Startup Area Control)

The FSPR bit indicates the protection state against the configuration set command for the BTFLG bit, and FSUACR Register.

In response to a reset or configuration set command, the FSCI transfers data from flash memory to this register.

### BTFLG bit (Flag of Startup Area Select for Boot Swap)

The BTFLG bit indicates whether the address of the startup area is exchanged for the boot swap function or not.

In response to a reset or configuration set command, the FSCI transfers data from flash memory to this register.

## 49.4.27 FCPSR : Flash Sequencer Processing Switching Register

Base address: FSCI = 0x4011\_E000  
FSCI\_NS = 0x5011\_E000

Offset address: 0xE0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ESUS PMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ESUSPMD	Erase Suspend Mode 0: Suspension priority mode 1: Erase priority mode.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

FCPSR selects the erasure suspension mode. FCPSR is initialized when the SUNIT bit in FSUNITR is set to 1. It is also initialized by a reset.

### ESUSPMD bit (Erase Suspend Mode)

The ESUSPMD bit selects the erasure suspension mode when a P/E suspend command is issued while the flash sequencer is executing erasure processing (see [section 49.9.3.10. P/E Suspend Command](#)). This bit should be set before issuing Block Erase or Multi Block Erase command.

### 49.4.28 FPCKAR : Flash Sequencer Processing Clock Notification Register

Base address: FACI = 0x4011\_E000  
FACI\_NS = 0x5011\_E000

Offset address: 0xE4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]							PCKA[7:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

Bit	Symbol	Function	R/W
7:0	PCKA[7:0]	Flash Sequencer Operating Clock Notification These bits are used to set the operating frequency of the flash sequencer while processing FACI commands.	R/W <sup>*1,2</sup>
15:8	KEY[7:0]	Key Code	W <sup>*3</sup>

Note: S-TYPE-3, P-TYPE-2

Note: Associated SA: FSAR register. See [section 49.4.4. FSAR : Flash Security Attribution Register](#)

Note 1. This bit can be written when the FRDY bit in the FSTATR register is 1. Writing to this bit is ignored when the FRDY bit is 0.

Note 2. Writing to these bits is only possible when 16 bits are written and the value written to the KEY[7:0] bits is 0x1E.

Note 3. Written values are not retained by these bits (always read 0x00).

FPCKAR specifies the operating frequency of the flash sequencer while processing FACI commands. The highest operating frequency for the given product is set as the initial value.

#### PCKA[7:0] bits (Flash Sequencer Operating Clock Notification)

The PCKA[7:0] bits specify the operating frequency of the flash sequencer while processing FACI commands. Set the desired frequency for these bits before issuing an FACI command. Specifically, convert the frequency in MHz to a binary number and set it for these bits.

Example:

Frequency is 35.9 MHz (PCKA = 0x24)

Round up the first decimal place of 35.9 MHz to a whole number (= 36) and convert it into a binary number.

If the value set in these bits is smaller than the actual operating frequency of the flash sequencer, the flash memory programming/erasure characteristics cannot be guaranteed. If the value set in these bits is greater than the actual operating frequency of the flash sequencer, the flash memory programming/erasure characteristics can be guaranteed but the FACI command processing time such as the time programming/erasure takes will increase. The minimum FACI command processing time is obtained when the operating frequency of the flash sequencer is the same as the PCKA value.

#### KEY[7:0] bits (Key Code)

The KEY[7:0] bits control writing permission to the PCKA bit.

### 49.4.29 FSUACR : Flash Startup Area Control Register

Base address: FACI = 0x4011\_E000

Offset address: 0xE8

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	KEY[7:0]							—	—	—	—	—	—	SAS[1:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



Bit	Symbol	Function	R/W
1:0	SAS[1:0]	Startup Area Select 0 0: Startup area is selected by BTFLG bit 0 1: Startup area is selected by BTFLG bit 1 0: Startup area is temporarily switched to the default area (block 0) 1 1: Startup area is temporarily switched to the alternate area (block 1).	R/W <sup>*1 *3</sup>
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code	W <sup>*2</sup>

Note: S-TYPE-6, P-TYPE-2

Note 1. Following described the write condition of these bits (these conditions are required at the same time).

1. Access size to this register is 16 bits
2. The value of KEY[7:0] is 0x66
3. The FSPR bit is 1.

Note 2. Written values are not retained by these bits (always read 0x00).

Note 3. Only Secure access can write to this register. Both Secure access and Non-secure read access are allowed. Non-secure write access is denied, but TrustZone access error is not generated.

FSUACR sets the startup area for the boot swap function. Do not use this register in dual mode (the DUALSEL.BANKMD[2:0] bits are 000b). In dual mode, starting up proceeds from startup area 0.

### SAS[1:0] bits (Startup Area Select)

The SAS[1:0] bits select the startup area. Three methods are available for changing the startup area.

### KEY[7:0] bits (Key Code)

The KEY[7:0] bits control writing permission to the SAS [1:0] bits.

## 49.4.30 FCKMHZ : Data Flash Access Frequency Register

Base address: FLAD = 0x4011\_C000  
FLAD\_NS = 0x5011\_C000

Offset address: 0x40

Bit position: 7 6 5 4 3 2 1 0

Bit field: FCKMHZ[7:0]

Value after reset: 0 0 1 1 1 1 0 0

Bit	Symbol	Function	R/W
7:0	FCKMHZ[7:0]	Data Flash Access Frequency Register These bits optimize the speed of reading the data flash memory.	R/W

Note: S-TYPE-3, P-TYPE-2

Note: If the security attribution is configured as Secure:

- Secure access and Non-secure read access are allowed
- Non-secure write access is ignored, and TrustZone access error is not generated.

If the security attribution is configured as Non-secure:

- Secure and Non-secure access are allowed.

This register optimizes the speed of reading the data flash memory.

Set the frequency of the peripheral module clock (FCLK) of internal peripheral bus which is the clock for access to the data flash memory, in MHz units. For example, 35.9 MHz should be rounded up and set the frequency to 36. Number of cycles required for access to the data flash memory are inserted according to the frequency. When changing the frequency of the FCLK, follow the procedure below to modify the value of the data flash access frequency register (FCKMHZ) in either of the following ways according to whether operation is at a lower frequency before or after the change.

- When changing the speed from low to high: Modify FCKMHZ. After confirming the change by reading FCKMHZ, change the frequency.
- When changing the speed from high to low: Change the frequency. After the frequency is changed, modify FCKMHZ.

## 49.5 Flash Cache

### 49.5.1 Feature of flash cache

The FCACHE (Flash Cache) speeds up read access from bus master to the flash memory. The FCACHE includes:

- FCACHE1, for CPU instruction fetches
- FCACHE2, for CPU operand access and access from DMAC/DTC
- FLPF, for the prefetch access in CPU instruction fetches

**Table 49.10 Flash Cache 1 (FCACHE1) overview**

Cache Target Region	Secure alias : 0x0200_0000 - 0x020F_FFFF Non-secure alias : 0x1200_0000 - 0x120F_FFFF
Target Bus Master	CPU instruction Fetch
Capacity	256 Bytes
Associativity	8WAY set associative
	128 bits/entry (128 bit aligned data), 2 entries/way
Access Cycle	Cache Hit : 0 wait Cache Miss : wait number of Flash Wait Cycle Register

**Table 49.11 Flash Cache 2 (FCACHE2) overview**

Cache Target Region	Secure alias : 0x0200_0000 - 0x020F_FFFF Non-secure alias : 0x1200_0000 - 0x120F_FFFF
Target Bus Master	CPU Operand Access and Access from other than CPU
Capacity	16 Bytes
Associativity	Full Associative
	128 bits/entry (128 bit aligned data), 1 entry
Access Cycle	Cache Hit : 0 wait Cache Miss : wait number of Flash Wait Cycle Register

**Table 49.12 Prefetch Buffer (FLPF) overview**

Cache Target Region	Secure alias : 0x0200_0000 - 0x020F_FFFF Non-secure alias : 0x1200_0000 - 0x120F_FFFF
Capacity	32 Bytes
Associativity	Full Associative
	128 bits/entry (128 bit aligned data), 2 entries
Request Address	Next address of previous CPU Instruction
Access Cycle	Cache Hit : 0 wait Cache Miss : wait number of Flash Wait Cycle Register

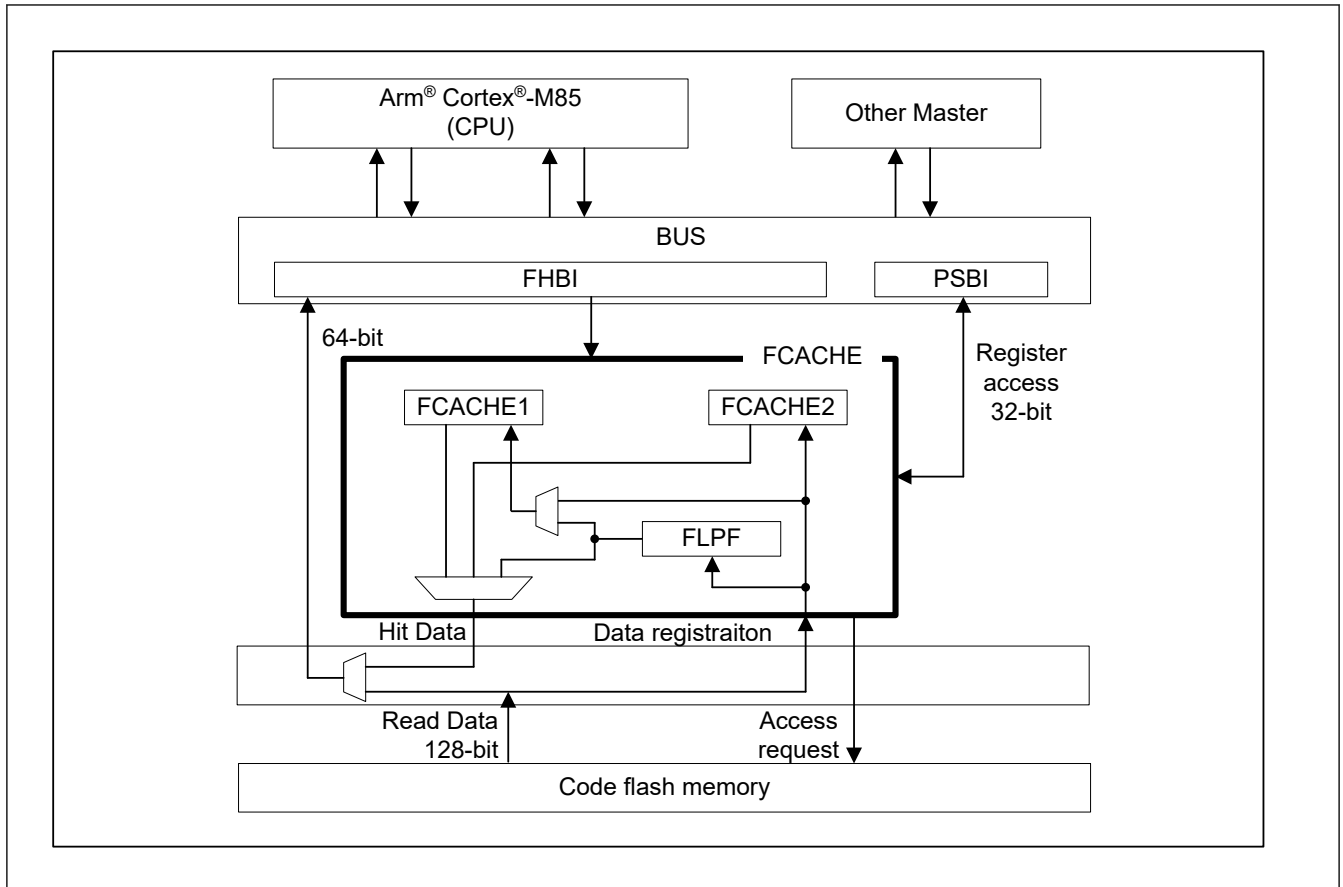


Figure 49.5 Block diagram of FCACHE

### 49.6 Operating Modes Associated with Flash Memory

Figure 49.6 is a diagram of the mode transitions associated with the flash memory. For the procedures for setting the modes, see section 6, Option-Setting Memory.

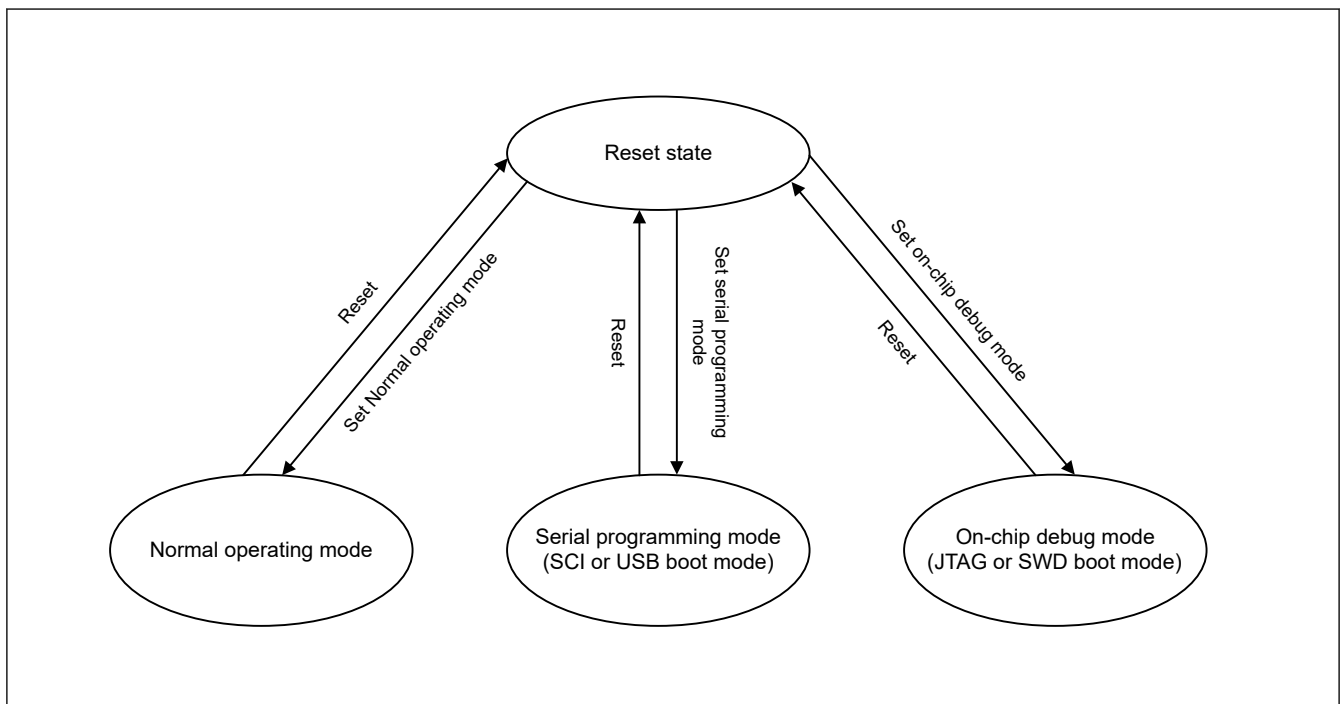


Figure 49.6 Mode Transitions Associated with Flash Memory

The flash memory area where programming and erasure are permitted and the boot program after a reset are different according to each mode. The differences between modes are listed in [Table 49.13](#).

**Table 49.13 Differences between Modes**

Parameter	Normal operating mode	Serial programming mode (SCI or USB boot mode)	On-chip debug mode (JTAG or SWD boot mode)
Programmable and erasable areas	<ul style="list-style-type: none"> <li>Code flash memory</li> <li>Data flash memory</li> <li>Option setting memory (programming only)</li> </ul>	<ul style="list-style-type: none"> <li>Code flash memory</li> <li>Data flash memory</li> <li>Option-setting memory (programming only)</li> </ul>	<ul style="list-style-type: none"> <li>Code flash memory</li> <li>Data flash memory</li> <li>Option setting memory (programming only)</li> </ul>
Erasure in block units	Possible	Possible	Possible
Boot program at a reset	User area program	Embedded program for serial programming	Depends on debug command

## 49.7 Overview of Functions

By using a dedicated flash-memory programmer to program the flash memory through a serial interface (serial programming) or JTAG/SWD interface (on-chip debug mode), the device can be rewritten regardless of whether this is before or after it is mounted on the target system.

Furthermore, security functions to prohibit rewriting or reading of the user program written to the flash memory are incorporated, and this can prevent falsification and illicit reading of the programs by third parties.

Programming by the user program (self-programming) is available to suit applications where the application on the target system may require updating after manufacturing or shipment. Protection features for the safe rewriting of the flash memory are also incorporated. Furthermore, interrupt processing during self-programming is supported, so programming can proceed at the same time as processing for external communications, etc., and this is the case in various situations. [Table 49.14](#) lists the overview of the methods of programming and the corresponding operating modes.

**Table 49.14 Programming methods**

Programming method	Functional overview	Operating mode
Serial programming	A dedicated flash-memory programmer through the SCI or USBFS interface enables on-board programming of the flash memory after the device is mounted on the target system.	Serial programming mode
	A dedicated flash-memory programmer through the SCI or USBFS interface and a dedicated programming adapter board allow off-board programming of the flash memory, for example, programming of the device before it is mounted on the target system.	
Self-programming	<p>A user program written to memory in advance of serial programming execution can also program the flash memory. The background operation capability makes it possible to fetch instructions or otherwise read data from the code flash memory while the data flash memory is programmed. As a result, a program resident in the code flash memory is able to program the data flash memory.</p> <p>For background operations that are not possible, instructions in the code flash memory cannot be fetched and data cannot be accessed while the code flash memory is being programmed by self-programming. In such cases, a program for programming from the internal SRAM or external memory must be transferred in advance and executed.</p>	Normal operating mode
JTAG or SWD programming	<p>A dedicated flash-memory programmer or an on-chip debugger through JTAG or SWD enables on-board programming of the flash memory after the device is mounted on the target system.</p> <p>A dedicated flash-memory programmer or an on-chip debugger through JTAG or SWD and a dedicated programming adapter board allow off-board programming of the flash memory, for example, programming of the device before it is mounted on the target system.</p>	On-chip debug mode

[Table 49.15](#) lists the functions of the flash memory. Serial programmer commands realize each function of serial programming, while reading of the flash memory by an FACI command or the user program realizes each function of self-programming.

**Table 49.15 Basic Functions**

Function	Functional overview	Availability	
		Serial programming	Self-programming
Blank check	Checks a specified block to ensure that writing to it has not already proceeded. Results of reading from data flash memory to which nothing is written after erasure are not guaranteed, so use blank checking to confirm that writing to memory has not proceeded after erasure.	Not supported	Supported (data flash programming only)
Block erasure	Erases the memory contents in the specified block	Supported	Supported
Programming	Writes to the specified address	Supported	Supported
CRC	Calculates the CRC in the specified range of the flash memory and transfers it to the flash programmer	Supported	Not supported
Read	Reads data programmed in the flash memory	Supported	Not supported (read by user program is possible)
Start-up program protection functions	Configures the start-up program protection functions	Supported	Supported
Option function selection	Selects the option function, and modifies the initial setting of this MCU	Supported	Supported
Dual bank function	Switches different modes (linear or dual)	Supported	Supported
Block swap function	Setting block swap functions	Supported	Supported
Block protection	Setting block protection	Supported	Supported
Device lifecycle transition	Transitions the device lifecycle	Supported	Not supported
Memory security attribution	Setting the memory security attribution	Supported	Not supported
Key	Injects key	Supported	Supported (except the key related to device lifecycle transition)
All erasure	Erase the flash memory to the state after shipment	Supported	Not supported

The flash memory supports various security functions.

[Table 49.16](#) lists the security functions supported by the flash memory.

**Table 49.16 Lists of Security Functions**

Function	Description
Security flag for Start-up area select	Start-up area selection can be protected by setting of security flag (FSPR).
Permanently block protection	Programming or erasure of each block of code flash memory can be protected permanently.
Protection for TrustZone	Programming or erasure area, readable area, register access, and FACI command operation are protected by ARM TrustZone security.
Data flash configuration area protection	When target area of configuration set command includes protection area by lock bit, flash sequencer detects an error and enters command lock state.
Anti-rollback Counter	Anti-rollback counter used only by Secure applications for firmware updates.
Programming or erasure mode protection	Only Secure developer can enter the programming or erasure mode for code flash.

## 49.8 Operating Modes of the Flash Sequencer

The flash sequencer has three operating modes as shown in [Figure 49.7](#). Transitions between modes are initiated by changing the value of the FENTRYR register.

When the value of the FENTRYR register is 0x0000, the flash sequencer is in read mode. In this mode, it does not receive FACI commands. The code flash memory and data flash memory are both readable.

When the value of the FENTRYR register is 0x0001, the flash sequencer is in code flash P/E mode where the code flash memory can be programmed or erased by FACI commands. In this mode, the data flash memory is readable. In addition, the code flash memory is not readable if background operation (BGO) is disabled. If BGO is enabled, the code flash memory

that is not selected by FSADDR register is readable when the FRDY bit in the FSTATR register is 0. As for the condition for enabling BGO, see [section 49.15.2. Background Operation](#).

When the value of the FENTRYR register is 0x0080, the flash sequencer is in data flash P/E mode where the data flash memory can be programmed or erased by FACI commands. In this mode, the data flash memory is not readable. However, the code flash memory is readable.

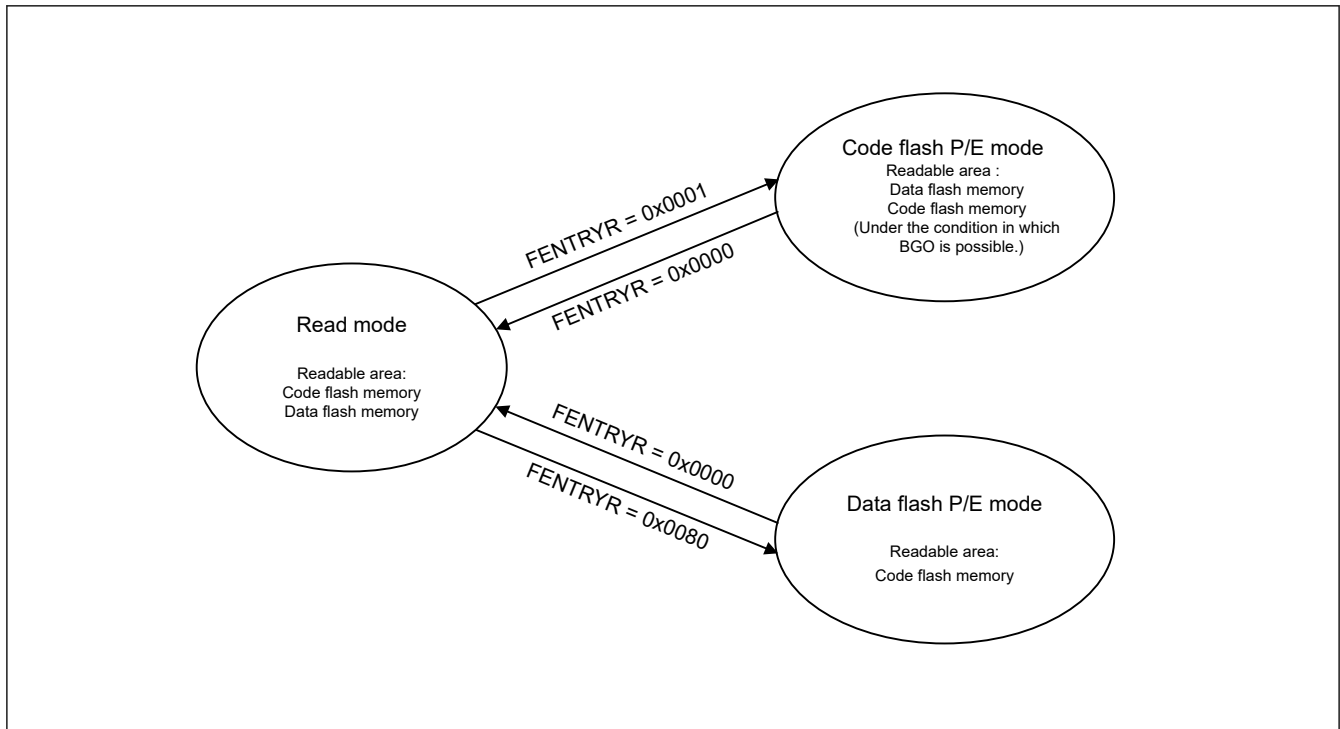


Figure 49.7 Modes of the flash sequencer

## 49.9 FACI Commands

### 49.9.1 List of FACI Commands

The FACI controls the FCU according to the specified FACI commands.

This section describes information about the FACI commands and [Table 49.17](#) lists the FACI commands.

Table 49.17 FACI commands (1 of 2)

FACI command	Function
Program	Programs the user area and data area. Units of programming are 128 bytes for the user area and 4, 8, or 16 bytes for the data area.
Block erase	Erases user area and data area. The erase unit is 8 KB or 32 KB for user area, and 64 bytes for data flash.
Multi block erase	Erases data area. The erase unit is 64, 128, 256 bytes for data flash.
P/E suspend	Suspends programming or erasure processing.
P/E resume	Resumes suspended programming or erasure processing.
Status clear	Initializes the IGLERR, ERSERR, PRGERR, ILGCOMERR, FESETERR, SECERR, and OTERR bits in the FSTATR register and the CMDLK, CFAE, OTERR and TZFERR bits in the FASTAT register, and the flash sequencer released from command-locked state.

**Table 49.17 FACI commands (2 of 2)**

FACI command	Function
Forced stop	Forcibly stops processing of FACI commands and initializes the FSTATR and FASTAT registers.
Blank check	Checks if data areas are blank. Units of Blank Check: 4 bytes to data flash memory capacity (specified in 4-byte units).
Configuration set	Sets the option-setting memory. Units of setting: <ul style="list-style-type: none"> <li>• 16 bytes (Code flash memory)</li> <li>• 4/16 bytes (Data flash memory)</li> </ul>
Increment Counter (Secure access only)	Increments Anti-rollback counter value . The target of counter is selected by FCNTSELR register. Update data unit is 1bit.
Refresh Counter recovery flow (Secure access only)	This is used to anti-rollback counter recovery flow. The target of counter is selected by FCNTSELR register. This command can refresh anti-rollback counter area without increasing counter value.
Read Counter (Secure access only)	Read Anti-rollback counter value. The target of counter is selected by FCNTSELR register. Reading data is output to FCNTDATAR0 and FCNTDATAR1 registers. Read unit is 64bit.

The FACI commands are issued by writing to the FACI command-issuing area (see [Table 49.4](#)). When write access as shown in [Table 49.18](#) proceeds in the specified state, the flash sequencer executes the processing associated with the given command (see [section 49.9.2. Relationship between the Flash Sequencer State and FACI Commands](#)).

**Table 49.18 FACI command formats (1 of 2)**

FACI commands	Number of write access	Write data to the FACI command-issuing area			
		1st access	2nd access	3rd to (N+2)th access	(N+3)th access
Program (user area) N = 64	67	0xE8	0x40 (=N)	WD1 to WD64	0xD0
Program (data area) 4-byte programming: N = 2 8-byte programming: N = 4 16-byte programming: N = 8	N+3	0xE8	0x02 (=N) 0x04 (=N) 0x08 (=N)	WD1 to WDN	0xD0
Block Erase (user area 8K/32K Bytes)	2	0x20	0xD0	—	—
Block Erase (data area 64 bytes)	2	0x20	0xD0	—	—
Multi block erase (data area 64/128/256 bytes)	2	0x21	0xD0	—	—
P/E suspend	1	0xB0	—	—	—
P/E resume	1	0xD0	—	—	—
Status Clear	1	0x50	—	—	—
Forced Stop	1	0xB3	—	—	—
Blank Check	2	0x71	0xD0	—	—
Configuration set (Code flash memory) N = 8	11	0x40	0x08 (=N)	WD1 to WD8	0xD0
Configuration set (Data flash memory) N = 2 (4 Bytes) N = 8 (16 Bytes)	N+3	0x40	0x02 (=N) 0x08 (=N)	WD1 to WDN	0xD0
Increment Counter	2	0x35	0xD0	—	—

**Table 49.18 FACI command formats (2 of 2)**

FACI commands	Number of write access	Write data to the FACI command-issuing area			
		1st access	2nd access	3rd to (N+2)th access	(N+3)th access
Refresh Counter	2	0x37	0xD0	—	—
Read Counter	2	0x39	0xD0	—	—

Note: WDN (N = 1, 2, ...): Nth 16-bit data to be programmed.

The flash sequencer clears the FSTATR.FRDY bit to 0 at the start of a command processing other than the Status Clear command, and sets this bit to 1 on completion.

If the FRDYIE.FRDYIE bit setting is 1, a flash ready (FRDY) interrupt is generated when the FSTATR.FRDY bit is set to 1.

## 49.9.2 Relationship between the Flash Sequencer State and FACI Commands

The FACI commands are accepted according to the mode/state of the flash sequencer. FACI commands should be issued after transitioning of the flash sequencer to the code flash P/E mode or data flash P/E mode and after checking the state of the flash sequencer.

Use the FSTATR and FASTAT registers to check the state of the flash sequencer. In addition, the occurrence of errors in general can be checked by reading the CMDLK bit in the FASTAT register. The value of the CMDLK bit is the logical OR of the following bits in the FSTATR register:

- ILGLERR
- ILGCOMERR
- FESETERR
- SECERR
- OTERR
- TZFERR
- ERSERR
- PRGERR
- FLWEERR.

Table 49.19 lists the available FACI commands in each operating mode.

**Table 49.19 Operating mode and available FACI commands**

Operating mode	FENTRYR	Available FACI commands
Read mode	0x0000	None
Code flash P/E mode	0x0001	Program Block erase P/E suspend P/E resume Status Clear Forced Stop Configuration set
Data flash P/E mode	0x0080	Program Block erase Multi block erase P/E suspend P/E resume Status Clear Forced Stop Blank Check Configuration set Increment Counter Refresh Counter Read Counter



Table 49.20 shows the state of the flash sequencer and acceptable FACI commands. An appropriate mode is assumed to have been set before the commands are executed.

**Table 49.20 Acceptable FACI commands and state of the flash sequencer**

	Program, block erase or multi block erase command processing	Configuration set, configuration clear, increment counter, refresh counter or read counter command processing	Program, block erase or multi block erase command suspension processing	Blank check command processing	Programming suspended	Erasure suspended	Programming while erasure is suspended	Command-locked state (FRDY = 1)	Command-locked state (FRDY = 0)	Processing of forced stop command	Other state
FRDY bit	0	0	0	0	1	1	0	1	0	0	1
SUSRDY bit	1	0	0	0	0	0	0	0	0	0	0
ERSSPD bit	0	0	0/1	0/1	0	1	1	0/1	0/1	0	0
PRGSPD bit	0	0	0/1	0/1	1	0	0	0/1	0/1	0	0
CMDLK bit	0	0	0	0	0	0	0	1	1	0	0
Program	X	X <sup>*3</sup>	X	X	X	O <sup>*2</sup>	X	X	X	X	O
Block erase or multi block erase	X	X <sup>*3</sup>	X	X	X	X	X	X	X	X	O
P/E suspend	O	X <sup>*3</sup>	X	X	X	X	X	—	X	X	—
P/E resume	X	X <sup>*3</sup>	X	X	O	O	X	X	X	X	X
Status clear	X	X <sup>*3</sup>	X	X	O	O	X	O	X	X	O
Forced stop	O	O <sup>*3</sup>	O	O	O	O	O	O	O	O	O
Blank check	X	X <sup>*3</sup>	X	X	O <sup>*1</sup>	O <sup>*1</sup>	X	X	X	X	O <sup>*1</sup>
Configuration set	X	X <sup>*3</sup>	X	X	X	X	X	X	X	X	O
Increment Counter	X	X <sup>*3</sup>	X	X	X	X	X	X	X	X	O <sup>*1</sup>
Refresh Counter	X	X <sup>*3</sup>	X	X	X	X	X	X	X	X	O <sup>*1</sup>
Read Counter	X	X <sup>*3</sup>	X	X	X	X	X	X	X	X	O <sup>*1</sup>

Note: O: Acceptable  
 X: Not acceptable (places the sequencer in the command-locked state)  
 —: Ignored

Note 1. Only acceptable in data flash P/E mode.

Note 2. Acceptable when programming area is other than erase suspending block.

Note 3. When configuration set is processing and when FSTATR.DBFULL bit is 1, do not issue this command.

### 49.9.3 Usage of FACI Commands

#### 49.9.3.1 Overview of Command Usage in Code Flash P/E Mode

Figure 49.8 show an overview of FACI command usage in code flash P/E mode. For the available commands in code flash P/E mode, see Table 49.19.

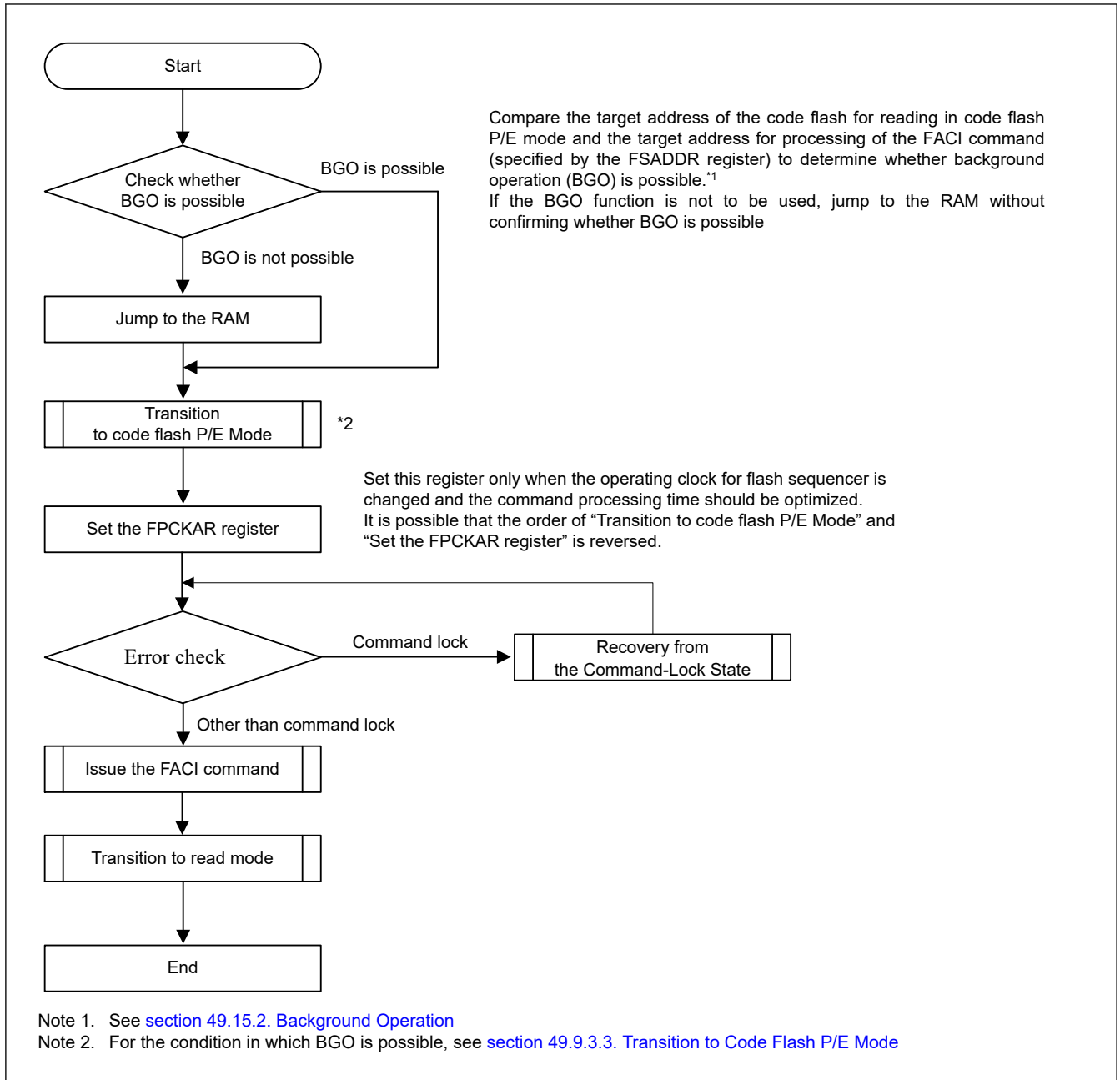
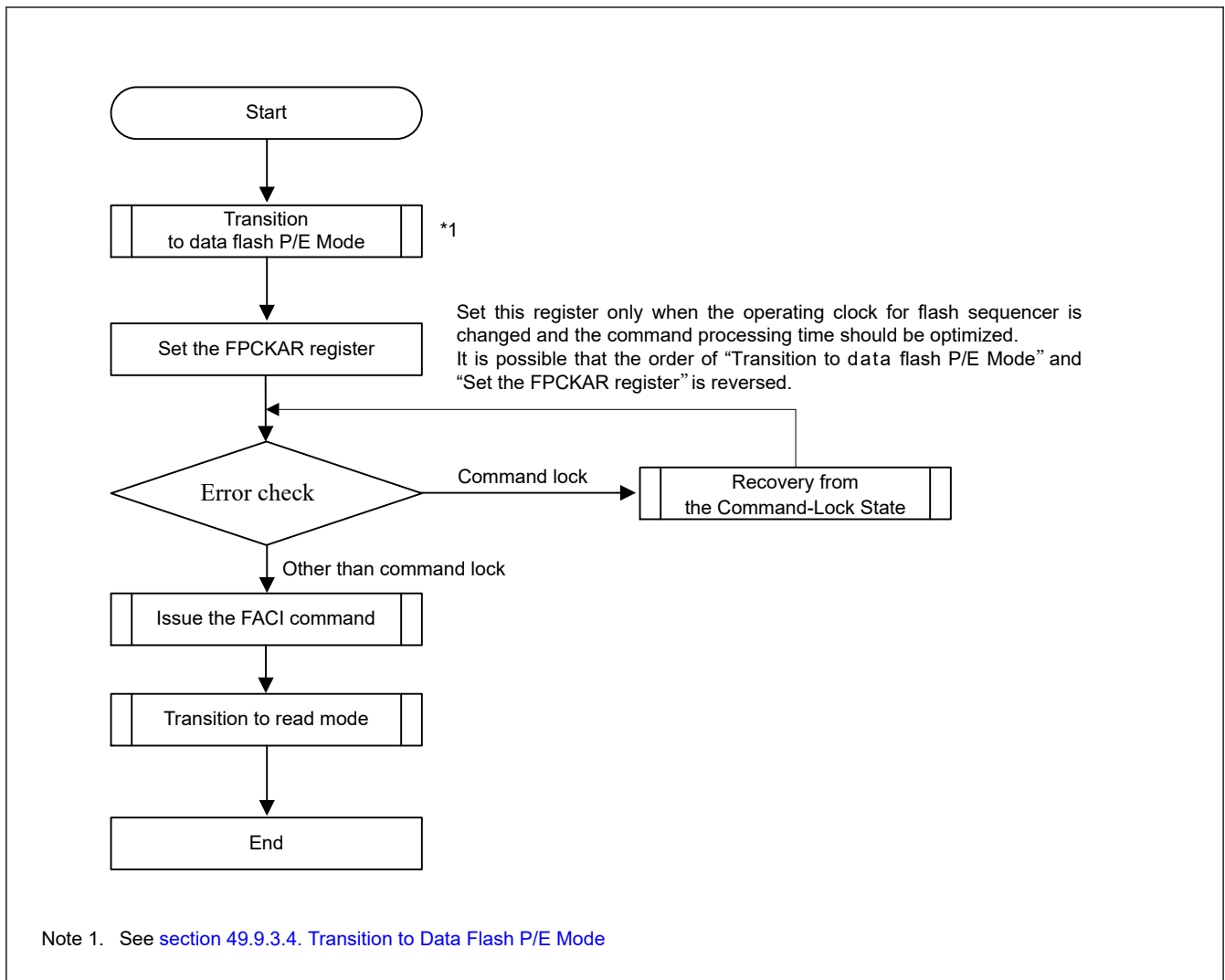


Figure 49.8 Overview of command usage in code flash P/E mode

### 49.9.3.2 Overview of Command Usage in Data Flash P/E Mode

Figure 49.9 shows an overview of FACL command usage in data flash P/E and Table 49.19 lists the available commands in data flash P/E mode.

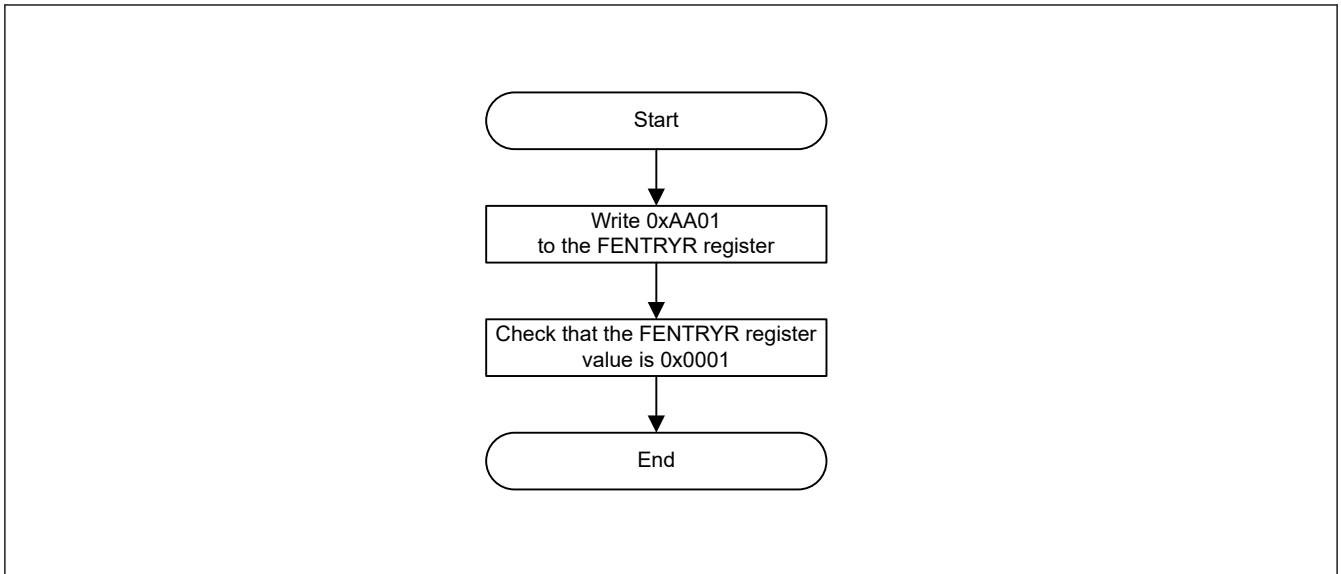


**Figure 49.9 Overview of command usage in data flash P/E mode**

### 49.9.3.3 Transition to Code Flash P/E Mode

To issue FACL commands for the code flash memory, a transition to code flash P/E mode is required by setting the FENTRYC bit in the FENTRYR register to 1.

[Figure 49.10](#) shows the procedure to transition to code flash P/E mode.

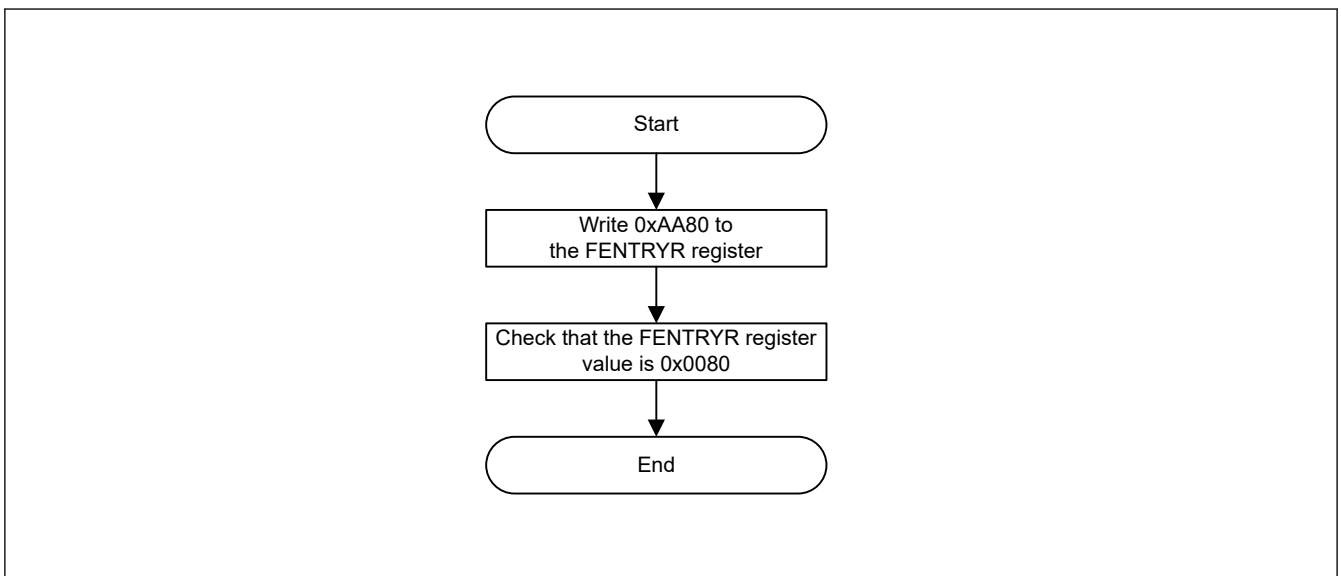


**Figure 49.10** Procedure to transition to code flash P/E mode

#### 49.9.3.4 Transition to Data Flash P/E Mode

To issue FACY commands for the data flash memory, a transition to data flash P/E mode is required by setting the FENTRYD bit in the FENTRYR register to 1.

Figure 49.11 shows the procedure to transition to data flash P/E mode.

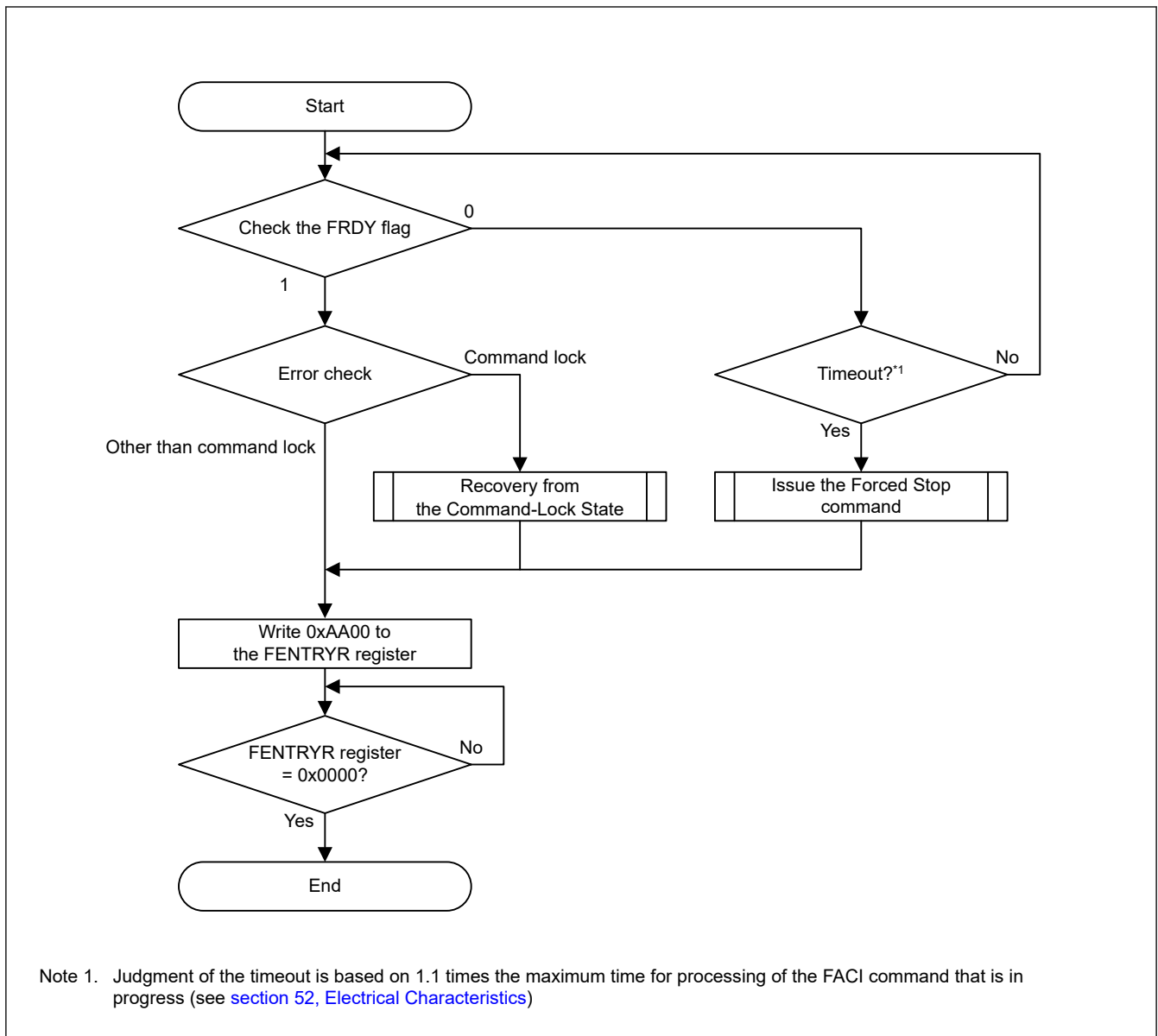


**Figure 49.11** Procedure to transition to data flash P/E mode

#### 49.9.3.5 Transition to Read Mode

To read the flash memory, a transition to read mode is required by setting the FENTRYR register to 0x0000. The transition to read mode should be made after the flash sequencer completes the processing and while operation is not in the command-locked state.

Figure 49.12 shows the procedure to transition to read mode.



**Figure 49.12 Procedure to transition to read mode**

### 49.9.3.6 Recovery from the Command-Locked State

When the flash sequencer enters the command-locked state, FACL commands cannot be accepted. To release the sequencer from the command-locked state, use the status clear command, forced stop command, or FASTAT register.

When the command-locked state is detected by checking for an error before issuing the P/E suspend command, the FRDY bit in the FSTATR register might be 0 even though command processing has not completed. If processing is not complete by the maximum programming/erasure time specified in the electrical characteristics, this is a timeout and the flash sequencer must be stopped with the forced stop command.

The FLWEERR bit in the FSTATR register does not change from 1 to 0 with the status clear command. When these bits are set to 1, use the forced stop command to release from the command-locked state. Bits other than FRDY and FLWEERR in FSTATR register that indicate the command-locked state can be changed from 1 to 0 with the status clear or forced stop command.

[Figure 49.13](#) shows the recovery flow from the command-locked state.

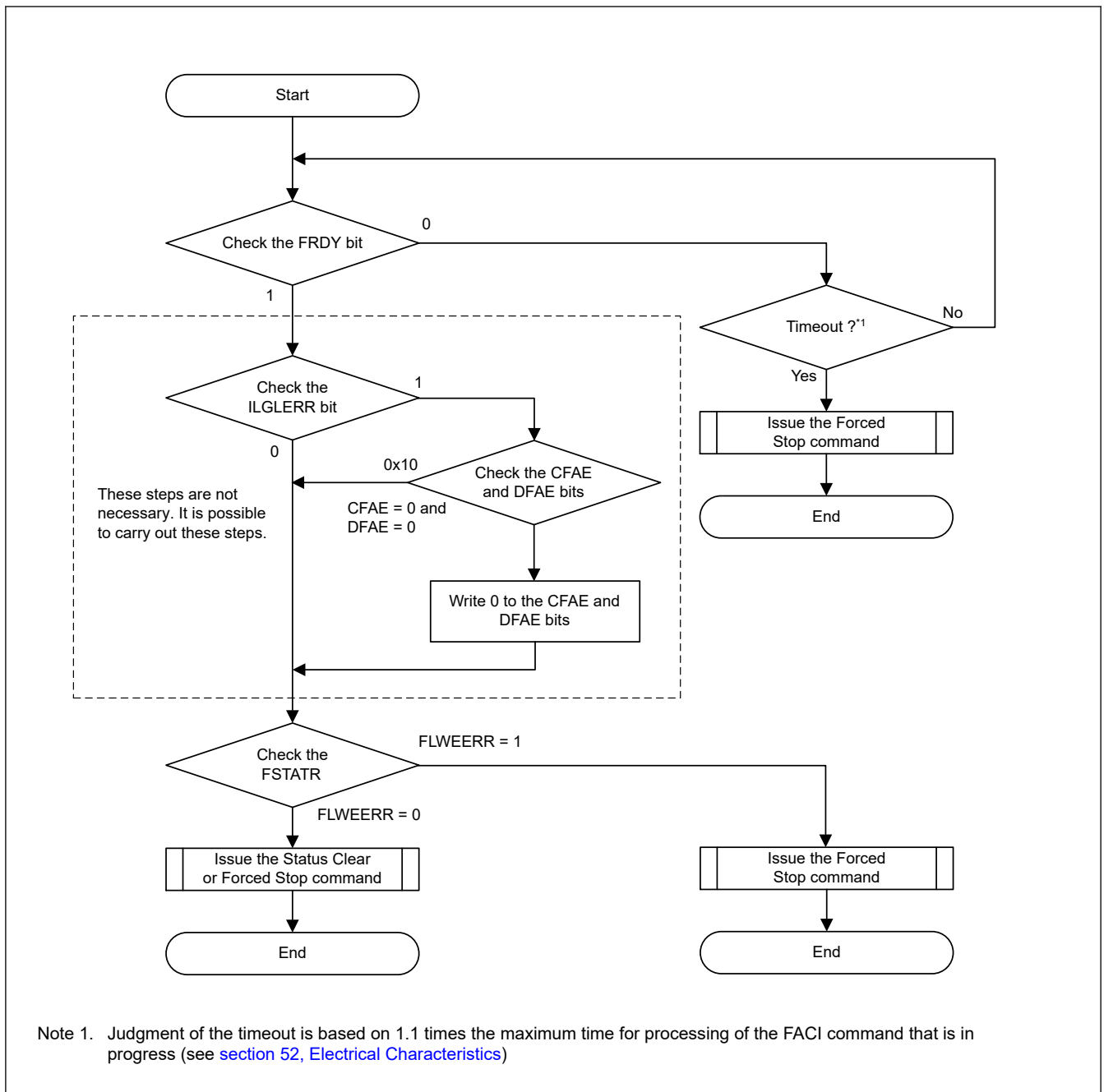


Figure 49.13 Recovery flow from the command-locked state

### 49.9.3.7 Program Command

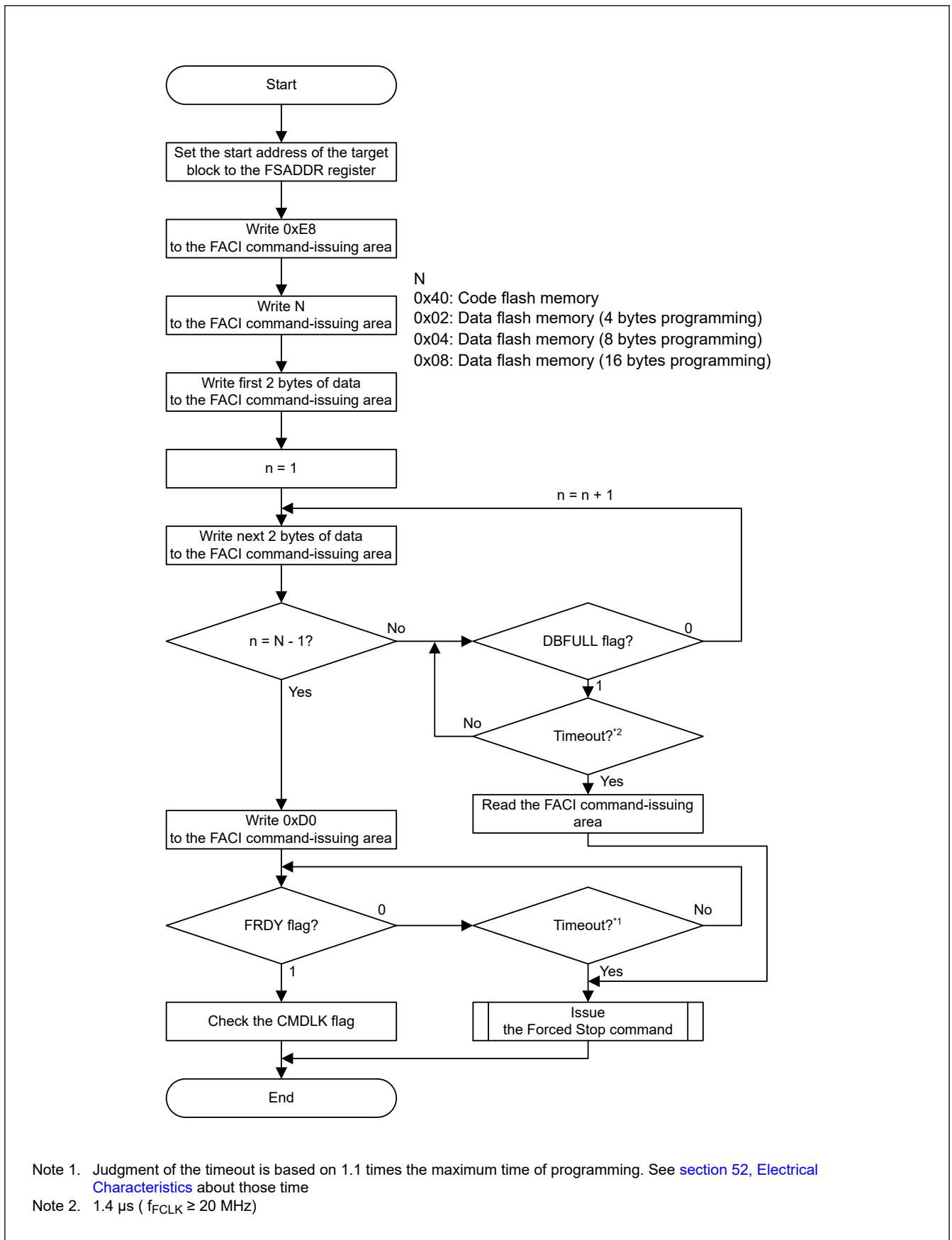
Program commands are used to write to the user and data areas. Before issuing the FACI program command, set the first address of the target block in the FSADDR register.

Program processing is started by writing the last 16 bits of program data (WD64 for code flash and WD2, WD4, and WD8 for data flash in Table 49.18) to the FACI command-issuing area before 0xD0, the final value of the FACI command, is written. Therefore, if it takes time from writing the last 16 bits of program data to writing the final FACI command value 0xD0 (for example, due to interrupt processing), the FSTATR.FR DY bit is set to 1 when program processing is completed even if 0xD0 has not been written. In this case, the program processing has been completed, but the FACI command reception has not been completed. If a subsequent FACI command is issued in this state, an illegal command error will occur. To avoid illegal command errors, disable interrupts with processing that issues FACI commands while program commands are issued (from writing 0xE8 to writing 0xD0).

If the target area for program command processing includes an area that is not for writing, write 0xFFFF to the corresponding area. If a program command is issued while the FACI internal data buffer is full, a wait period may occur on

the peripheral bus, which may affect the communication performance of other peripheral modules. To avoid the occurrence of a waiting period, set the DBFULL bit in the FSTATR register to 0 when issuing a FOCI command. Writing to the data area will not cause the data buffer to become full.

[Figure 49.14](#) shows the usage of the program command.



Note 1. Judgment of the timeout is based on 1.1 times the maximum time of programming. See [section 52, Electrical Characteristics](#) about those time

Note 2. 1.4 μs ( f<sub>CLK</sub> ≥ 20 MHz)

Figure 49.14 Usage flow of the program command



### 49.9.3.8 Block Erase Command

The block erase command is used for erasing user area or data area. The erase unit is one block. Before issuing a block erase command, set the first address of the target block to FSADDR register. Writing 0xD0 at the second write access of the FACL command triggers the FACL to start the block erase command processing. Completion of command processing can be confirmed with the FRDY bit of FSTATR register.

Set the FCPSR registers before issuing the block erase command. Additionally, FCPSR must be set when the erasure-suspended mode is to be switched.

Figure 49.15 shows the usage of the block erase command.

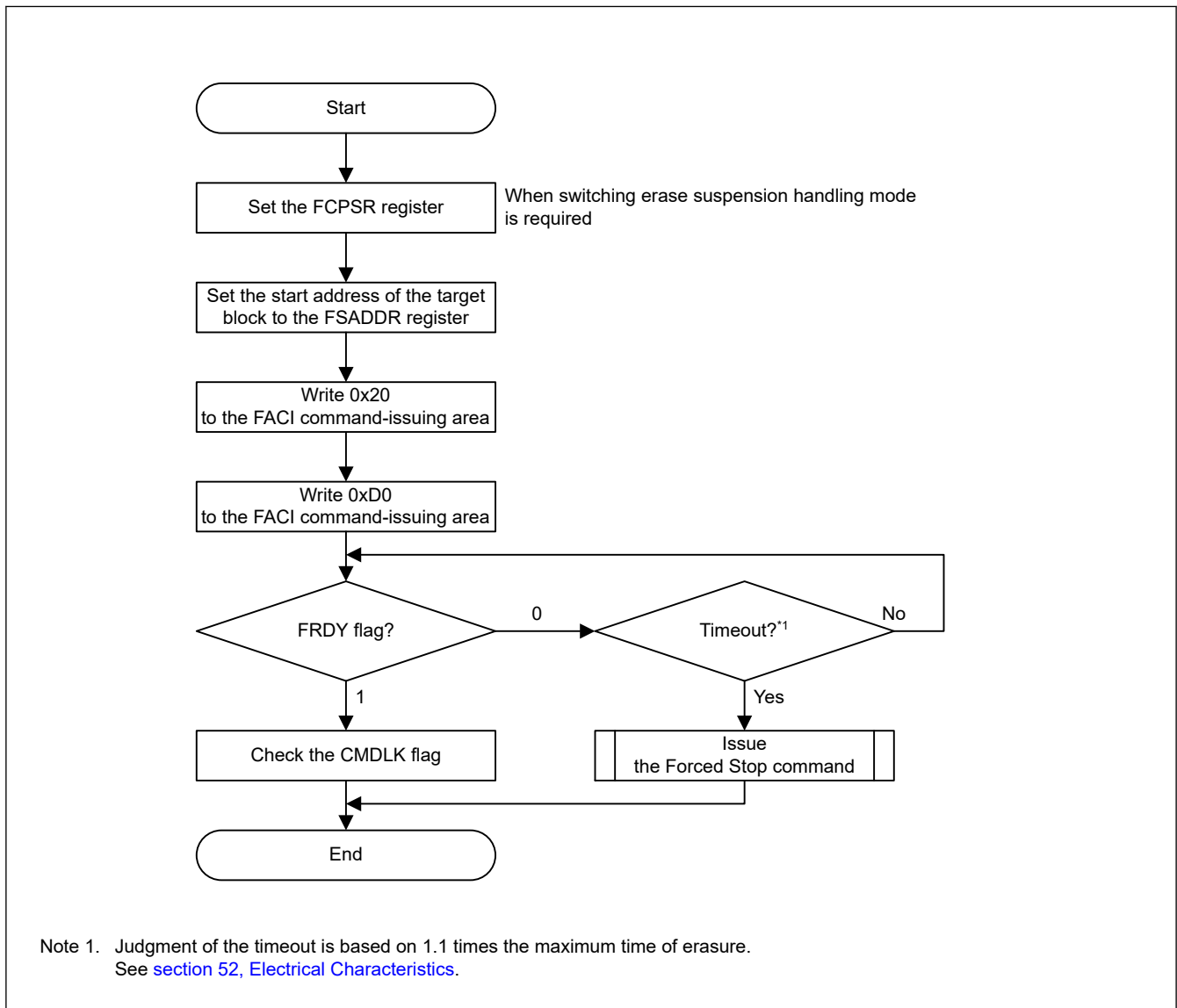


Figure 49.15 Usage flow of the block erase command

### 49.9.3.9 Multi Block Erase Command

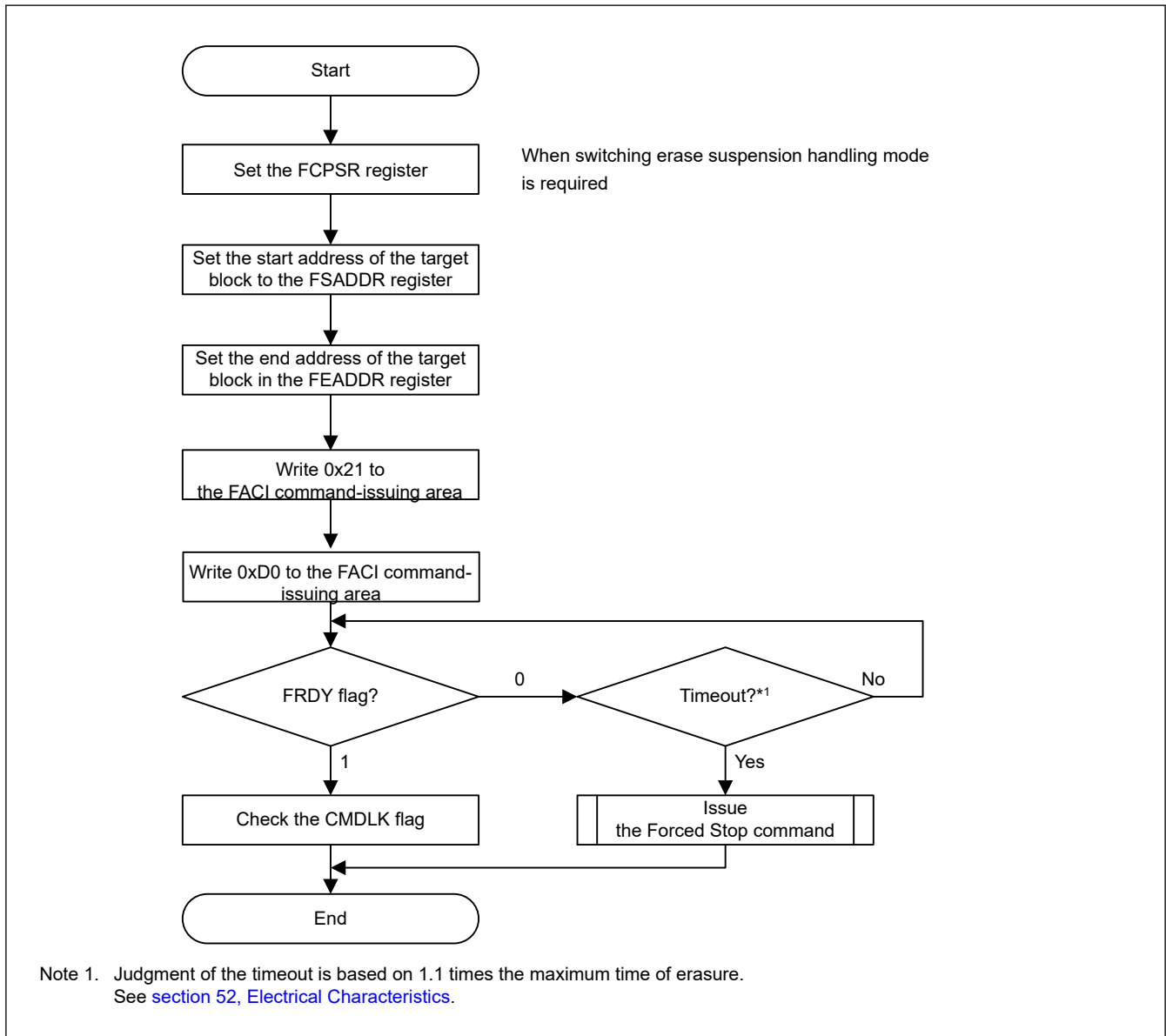
The multi block erase command is used for erasing data area. The erase unit is 64, 128, or 256 bytes. Before issuing the multi block erase command, set the start address to FSADDR register and the end address to FEADDR register. Writing 0xD0 at the second write access of the FACL command triggers FACL to start the multi block erase command processing. Completion of command processing can be confirmed with the FRDY bit of FSTATR register.

Set the FCPSR registers before issuing the multi block erase command. Additionally, FCPSR must be set when the erasure-suspended mode is to be switched.

The erase size is specified by both the FSADDR and FEADDR settings. [Table 49.21](#) describes how to set the FSADDR and FEADDR.

**Table 49.21 Settings for the erase size**

Erase size	FSADDR	FEADDR
64 bytes	FSA0 to FSA5 = 0 (64 byte-boundary)	FSADDR + 0x3C
128 bytes	FSA0 to FSA6 = 0 (128 byte-boundary)	FSADDR + 0x7C
256 bytes	FSA0 to FSA7 = 0 (256 byte-boundary)	FSADDR + 0xFC



**Figure 49.16 Usage flow of the multi block command**

### 49.9.3.10 P/E Suspend Command

The P/E suspend command is used to suspend programming/erasure. Before issuing a P/E suspend command, check that the CMDLK bit in the FASTAT register is 0, and that the execution of programming/erasure is performed normally. To confirm that the P/E suspend command can be received, check that the SUSRDY bit in the FSTATR register is 1. After issuing a P/E suspend command, read the CMDLK bit to confirm that no error occurs.

If an error occurs during programming/erasure, the CMDLK bit is set to 1. When programming/erasure processing has finished from the time when the SUSRDY bit is 1 to when the P/E suspend command is received, no error occurs and the

suspended state is not entered (the FRDY bit in the FSTATR register is 1 and the ERSSPD and PRGSPD bits in FSTATR are 0).

When a P/E suspend command is received and the programming/erasure suspend processing finishes normally, the flash sequencer enters the suspended state, the FRDY bit is set to 1, and the ERSSPD or PRGSPD bit is 1. After issuing a P/E suspend command, check that the ERSSPD or PRGSPD bit is 1 and the suspended state is entered, then proceed with the subsequent flow. If a P/E resume command is issued in the subsequent flow even when the suspended state is not entered, an illegal command error occurs and the flash sequencer shifts to the command-locked state (see [section 49.11.2. Error Protection](#)).

If the erasure suspended state is entered, programming to blocks other than an erasure target block can be performed. Additionally, the programming and erasure suspended states can shift to read mode by clearing the FENTRYR register.

[Figure 49.17](#) shows the usage of the P/E suspend command.

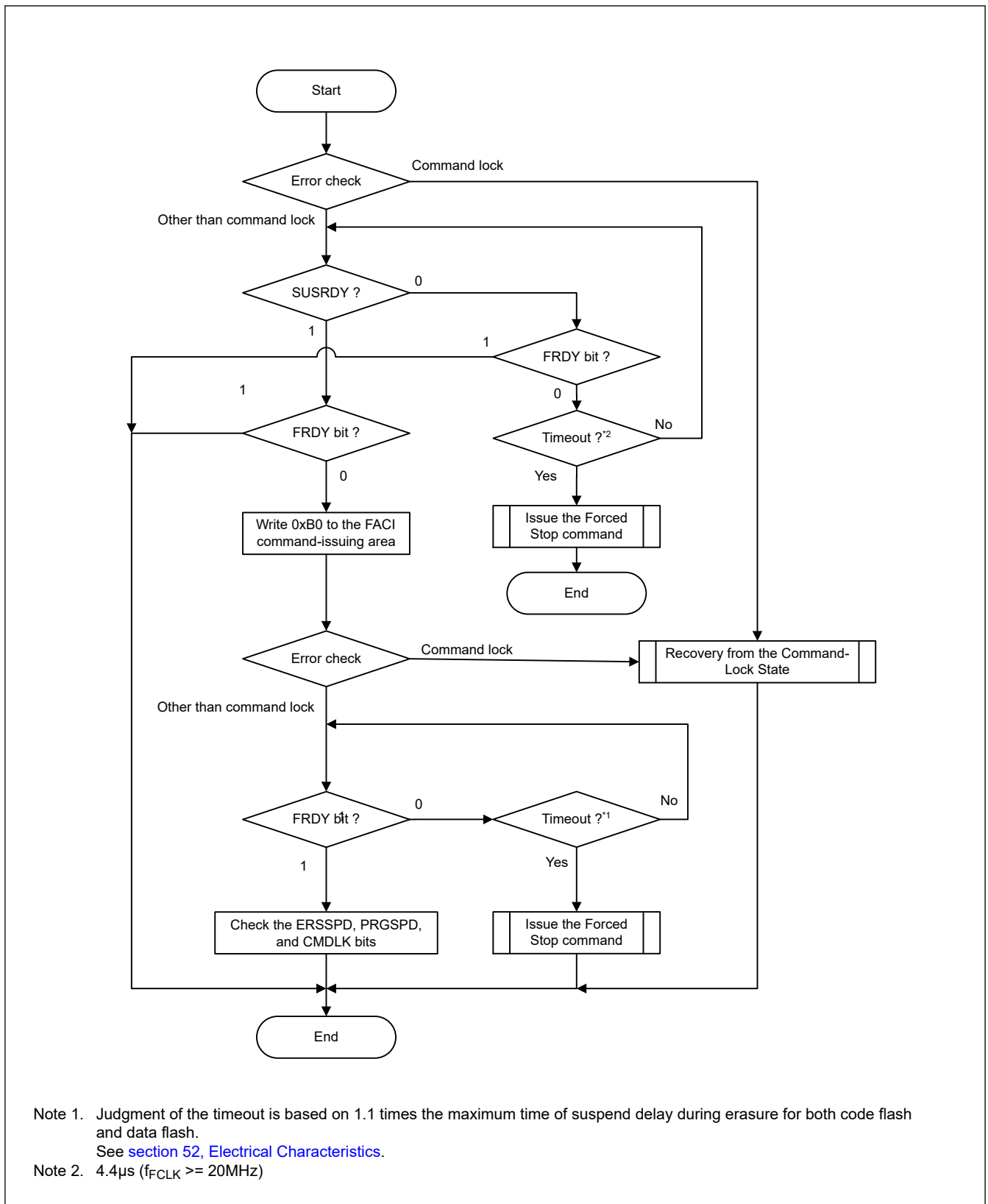


Figure 49.17 Usage flow of the P/E suspend command

(1) Suspension during Programming

When issuing a P/E suspend command during flash memory programming, the flash sequencer suspends programming processing. Figure 49.18 shows the suspend programming operation. When receiving programming-related command, the flash sequencer clears the FRDY bit in the FSTATR register to 0 to start programming. If the flash sequencer enters the

state in which the P/E suspend command can be received after programming starts, it sets the SUSRDY bit in the FSTATR register to 1.

When a P/E suspend command is issued, the flash sequencer receives the command and clears the SUSRDY bit to 0. If the flash sequencer receives a P/E suspend command while a programming pulse is applied, the flash sequencer continues with the pulse. After the specified pulse application time, the flash sequencer finishes pulse application, starts the programming suspend processing, and sets the PRGSPD bit in the FSTATR register to 1.

When a suspended processing finishes, the flash sequencer sets the FRDY bit to 1 to enter the programming suspended state. When receiving a P/E resume command in the programming suspended state, the flash sequencer clears the FRDY and PRGSPD bits to 0 and resumes programming.

Figure 49.18 shows the timing for suspension during programming.

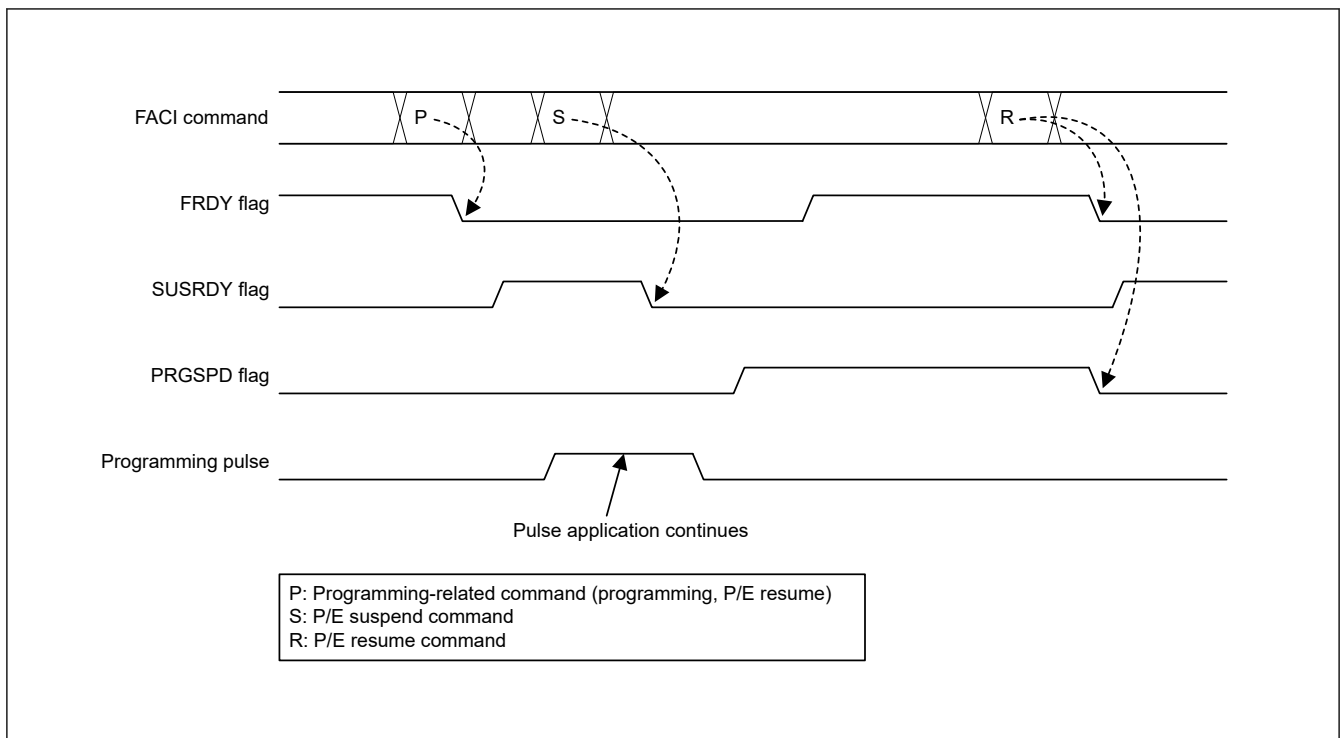


Figure 49.18 Suspension during programming

## (2) Suspension during Erasure (Suspension Priority Mode)

The flash sequencer has a suspension priority mode for the suspension of erasure. Figure 49.19 shows the suspend operation of erasure when the erasure suspend mode is set to the suspension priority mode (the ESUSPMD bit in the FCPSR register is 0).

When receiving an erasure-related command, the flash sequencer clears the FRDY bit in the FSTATR register to 0 to start erasure. If the flash sequencer enters the state in which the P/E suspend command can be received after erasure starts, it sets the SUSRDY bit in the FSTATR register to 1.

When a P/E suspend command is issued, the flash sequencer receives the command and clears the SUSRDY bit to 0.

When receiving a suspend command during erasure, the flash sequencer starts the suspend processing and sets the ERSSPD bit in the FSTATR register to 1 even when it is applying an erasure pulse. When the suspended processing finishes, the flash sequencer sets the FRDY bit to 1 to enter the erasure suspended state. When receiving a P/E resume command in the erasure suspended state, the flash sequencer clears the FRDY and ERSSPD bits to 0 and resumes erasure. Operations of the FRDY, SUSRDY, and ERSSPD bits at the suspension and resumption of erasure are the same, regardless of the erasure suspend mode.

The setting of the erasure suspend mode affects the control method of erasure pulses. In suspension priority mode, when receiving a P/E suspend command while erasure pulse A that has not been previously suspended is being applied, the flash sequencer suspends the application of erasure pulse A and enters the erasure suspended state. When receiving a P/E suspend command while reapplying erasure pulse A after erasure is resumed with a P/E resume command, the flash sequencer

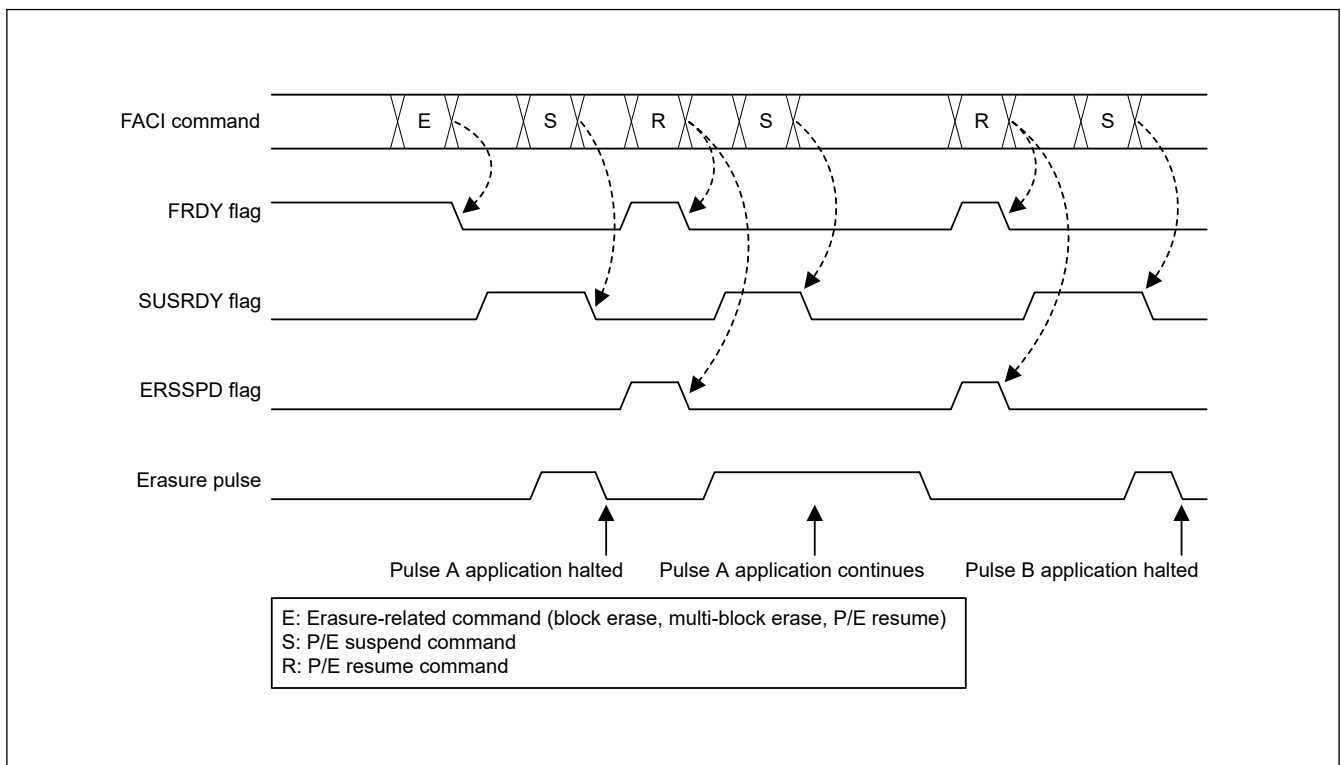
continues to apply erasure pulse A. After the specified pulse application time, the flash sequencer finishes erasure pulse application and enters the erasure suspended state.

When the flash sequencer receives a P/E resume command next and erasure pulse B is being applied, the flash sequencer receives a P/E suspend command again, and the application of erasure pulse B is then suspended. In suspension priority mode, delays due to suspension can be minimized because the application of an erasure pulse is suspended once per pulse, and priority is given to the suspend processing.

If the interval of suspension after resume is longer than  $t_{REST1}$  (Resume time: priority on suspension, resume after the 1st suspend for the same pulse), suspend delay will be always  $t_{SESD1}$  (Suspend delay: priority on suspension, the 1st suspend for the same pulse).

If the interval of suspension after resume is shorter than  $t_{REST1}$ , suspend delay becomes either  $t_{SESD1}$  or  $t_{SESD2}$  (Suspend delay: priority on suspension, the 2nd suspend for the same pulse).

(The value of  $t_{REST1}$  /  $t_{SESD1}$  /  $t_{SESD2}$ , see [section 52, Electrical Characteristics](#).)



**Figure 49.19 Suspension during erasure (suspension priority mode)**

### (3) Suspension during Erasure (Erasure Priority Mode)

The flash sequencer has an erasure priority mode for the suspension of erasure. [Figure 49.20](#) shows the suspend operation of erasure when the erasure suspend mode is set to the erasure priority mode (the ESUSPMD bit in the FCPSR register is 1). The control method of erasure pulses in erasure priority mode is the same as that of programming pulses for the programming suspend processing.

If the flash sequencer receives a P/E suspend command while an erasure pulse is applied, the flash sequencer continues to apply the pulse. In this mode, the required time for the erasure processing can be reduced compared to the suspension priority mode because the re-application of erasure pulses does not occur when a P/E resume command is issued.

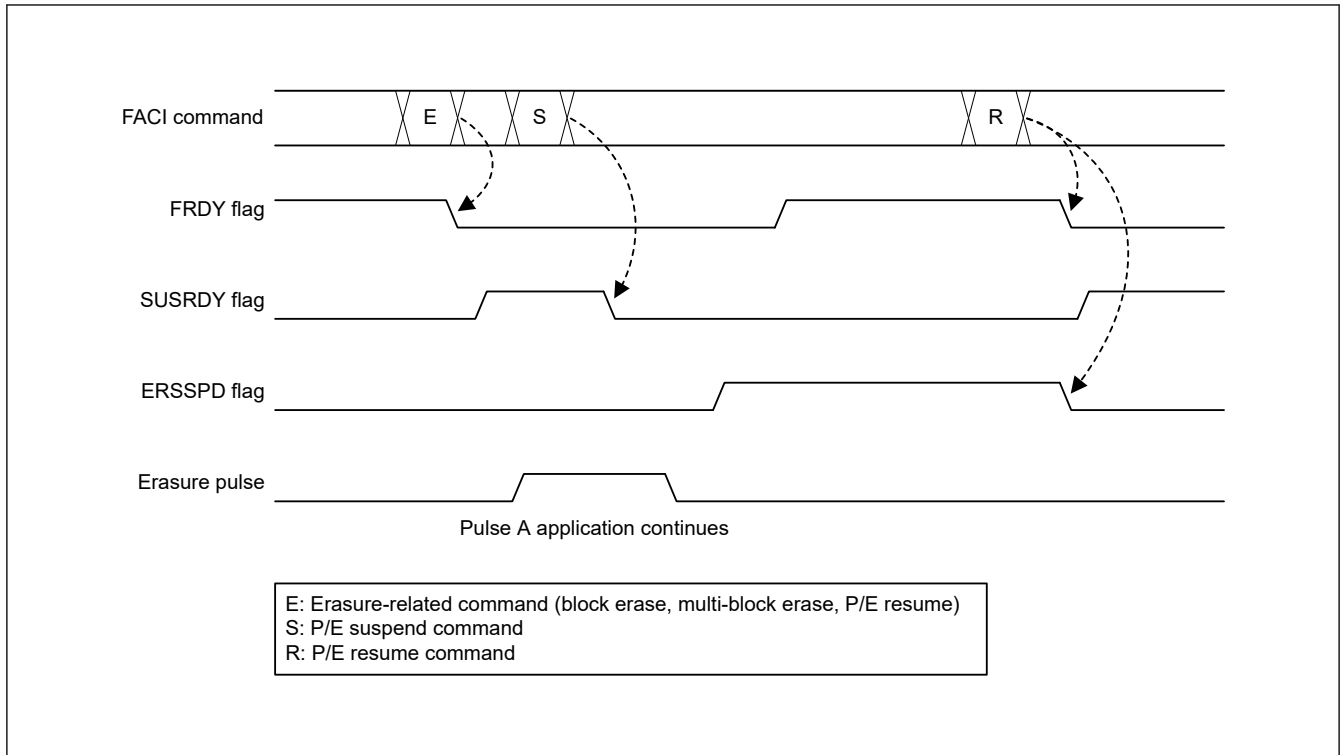


Figure 49.20 Suspension during erasure (erasure priority mode)

### 49.9.3.11 P/E Resume Command

The P/E resume command is used to resume suspended programming or erasure. If the FENTRYR setting has been modified during suspension, issue a P/E resume command only after resetting FENTRYR to the previous value that was held before the P/E suspend command was issued. Figure 49.21 shows usage of the P/E resume command.

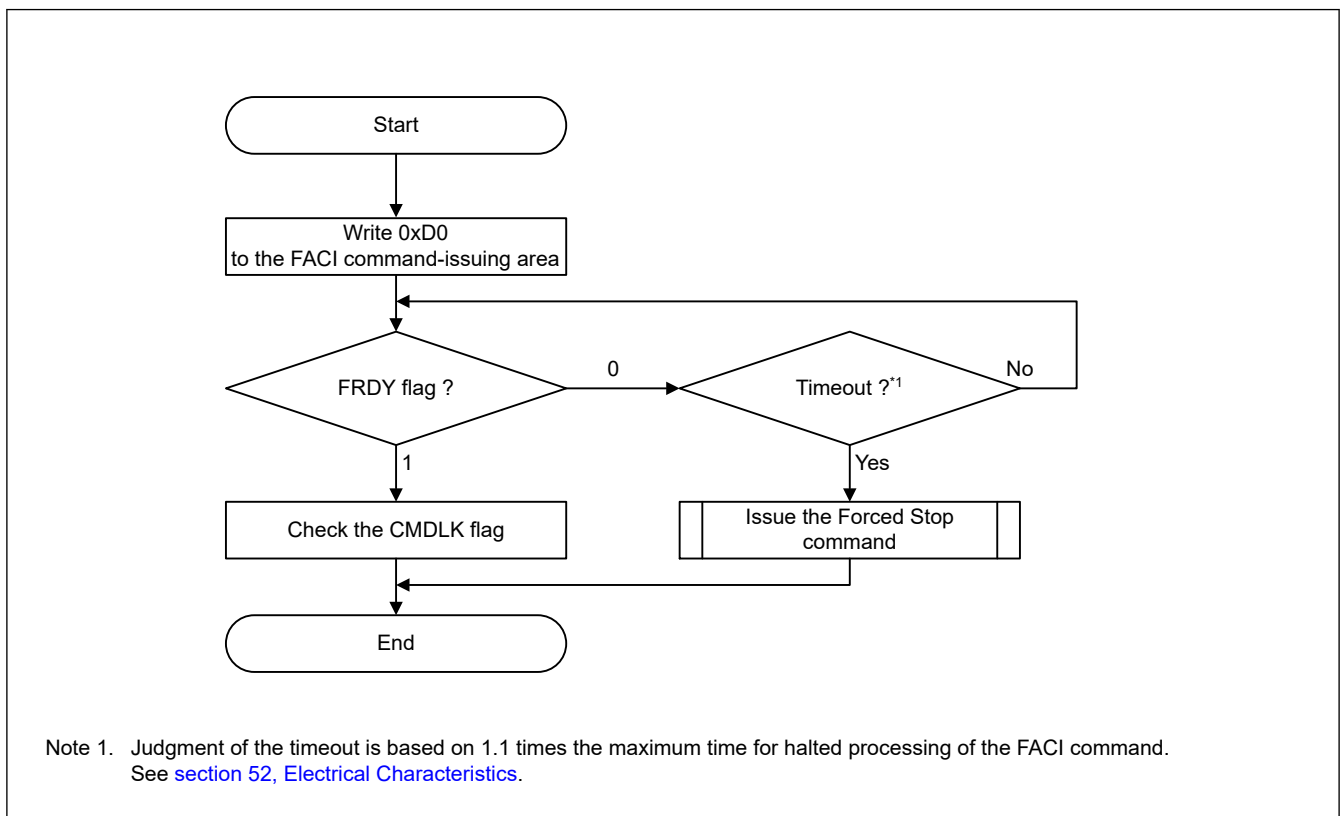


Figure 49.21 Usage flow of the P/E resume command

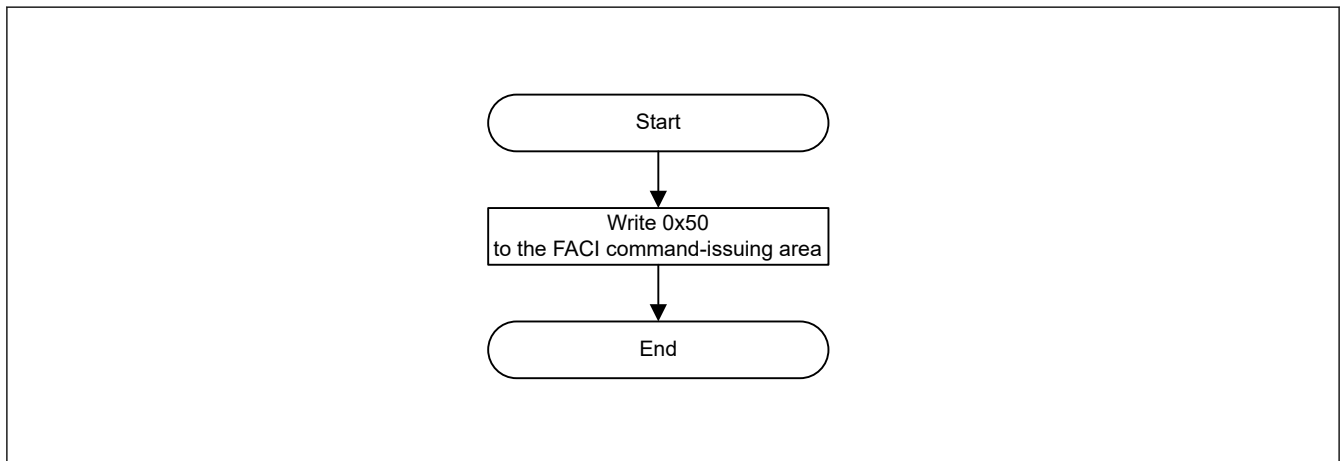
### 49.9.3.12 Status Clear Command

The status clear command is used to clear the command-locked state (see [section 49.9.3.6. Recovery from the Command-Locked State](#)).

You can use the status clear command to clear the following bits in the FSTATR register in the command-locked state:

- ILGLERR
- ILGCOMERR
- FESETERR
- SECERR
- OTERR
- ERSERR
- PRGERR
- TZFERR

[Figure 49.22](#) shows usage of the status clear command.



**Figure 49.22 Usage flow of the status clear command**

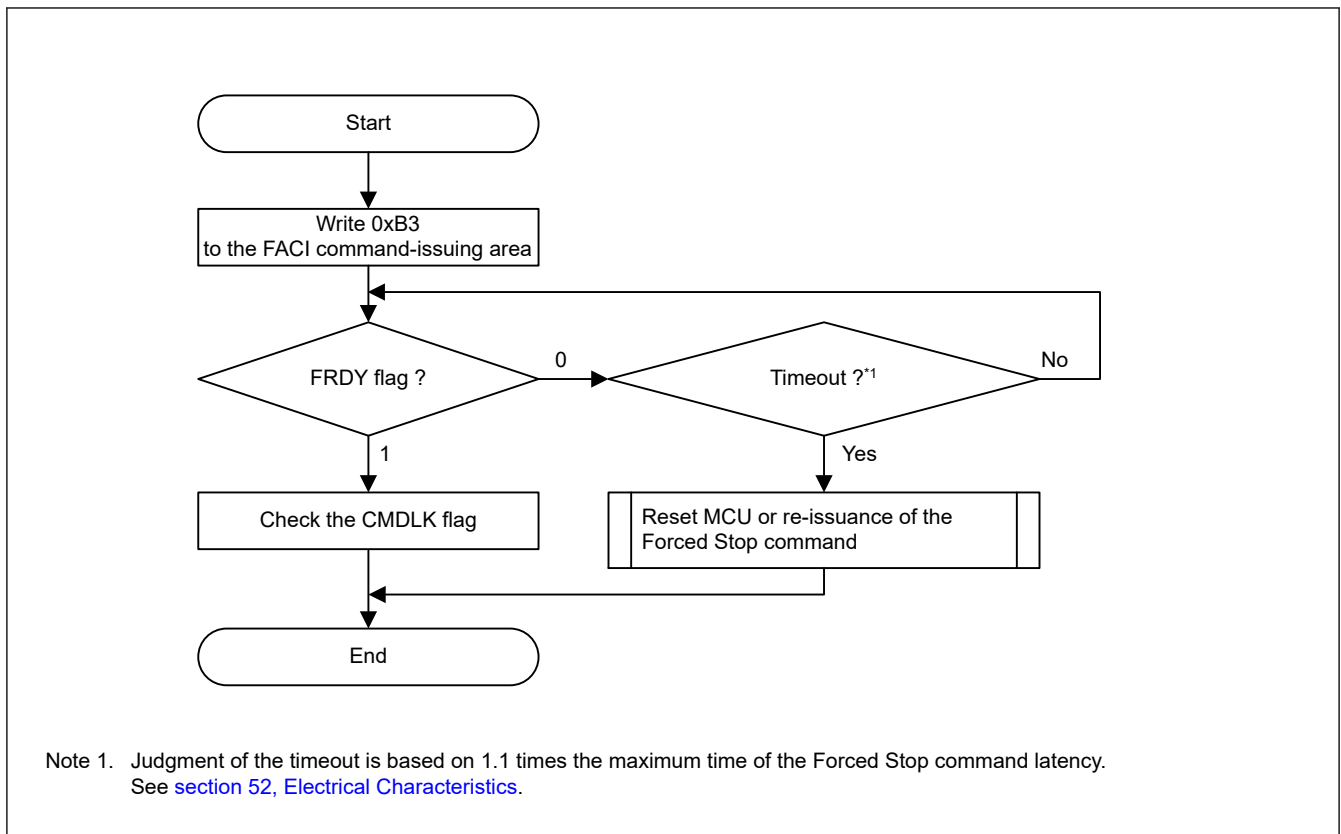
### 49.9.3.13 Forced Stop Command

The forced stop command is used to forcibly end command processing by the flash sequencer. Although this command halts command processing more quickly than the P/E suspension command, values from the programming or erasure that are in progress are not guaranteed. Additionally, resumption of processing is not possible. Processing of programming or erasure that is halted by the forced stop command is also defined as one programming round.

Executing the forced stop command also initializes part of the FACL, the whole FCU, the FSTATR and FASTAT registers. This command can be used in the procedure for recovery from the command-locked state and for processing in response to a timeout of the flash sequencer (see [section 49.9.3.6. Recovery from the Command-Locked State](#)).

[Figure 49.23](#) shows usage of the forced stop command.





**Figure 49.23 Usage flow of the forced stop command**

#### (1) Notes on Using the Forced Stop Command during Command Issue

When using the forced stop command at the timeout occurrence by DBFULL bit of the program command, writing in the FACI command-issuing area is sometimes processed as writing in data of the program command. See [Table 49.4](#) in [section 49.3. Address Space](#) for information on the FACI command-issuing area to force a command lock. Then issue a forced stop command with return method from the command lock status (see [Figure 49.14](#)). Locking commands is possible in any case where the unit for reading the FACI command issuing area is 8, 16, or 32 bits.

#### 49.9.3.14 Blank Check Command

The blank check command is used to confirm that an area is in the non-programmed state. Values read from the data flash memory that have been erased but not yet programmed again that is in the non-programmed state, are undefined.

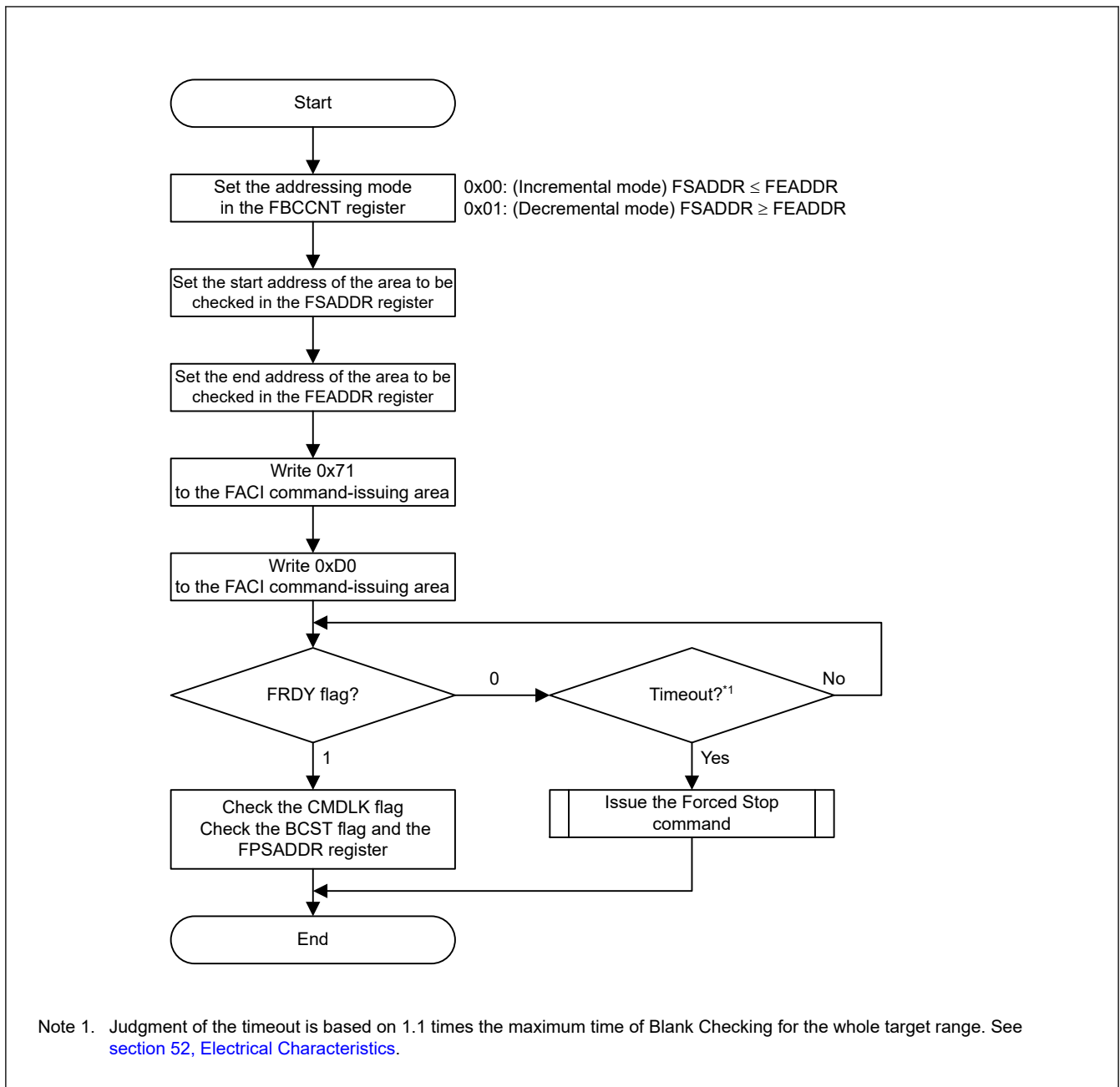
Before issuing the Blank Check command, set addressing mode, start address, and end address of the target area for Blank Check to the FBCCNT, FSADDR, and FEADDR registers. When Blank Check addressing mode is set to decremental mode (i.e. FBCCNT.BCDIR = 1), address specified in FSADDR should be equal to or larger than address in FEADDR.

On the other hand, the address in FSADDR should be equal to or smaller than address in FEADDR when Blank Check addressing mode is set to incremental mode (i.e. FBCCNT.BCDIR = 0).

If the settings of the BCDIR bit, FSADDR, and FEADDR are inconsistent, the flash sequencer enters the command-locked state. The size of the target area for Blank Check is in the range from 4 bytes to the data flash memory capacity and is set in units of 4 bytes.

Write 0x71 and 0xD0 to the FACI command-issuing area to start Blank Check. Completion of processing can be confirmed by the FRDY bit of the FSTATR register. At the end of processing, the result of Blank Check is stored in the BCST bit in the FBCSTAT register. If non-programmed data exists within the target area for Blank Check, flash sequencer stops Blank Check command operation. In this case, address of non-programmed data is indicated to FPSADDR register.

[Figure 49.24](#) shows usage of the blank check command.



**Figure 49.24 Usage flow of the blank check command**

### 49.9.3.15 Configuration Set Command

The Configuration set command is used to set option-setting memory. Before issuing the Configuration set command, set the specified address (shown in [Table 49.22](#)) in the FSADDR register. Writing 0xD0 to the FACI command-issuing area in the final access for issuing the FACI command starts FACI processing of the Configuration set command.

[Figure 49.25](#) shows usage of the configuration set command.

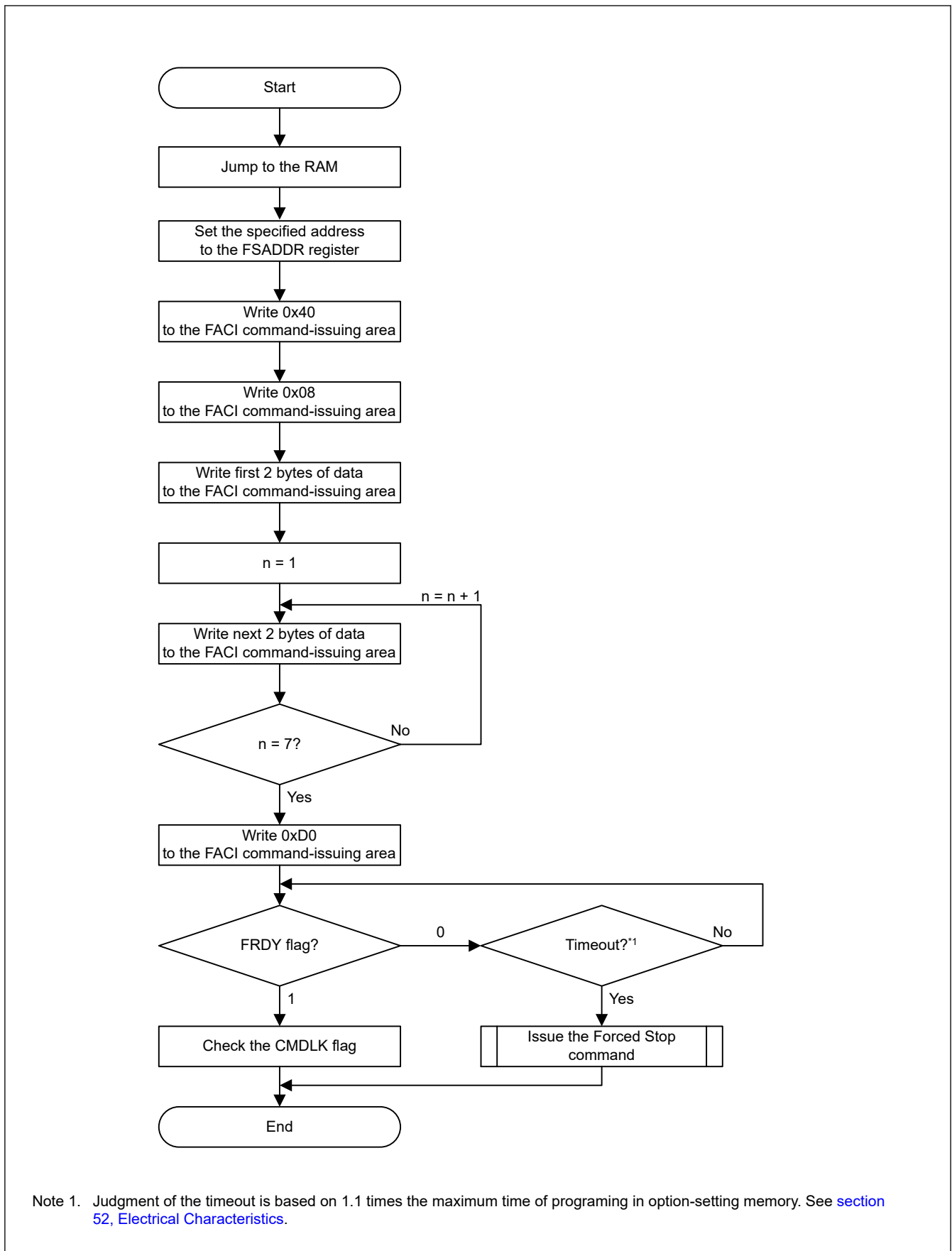


Figure 49.25 Usage flow of the configuration set command

The correspondence between the possible target data for configuration setting and the address value set in the FSADDR register is shown in Table 49.22. For details, see section 49.4.12. FSADDR : FACS Command Start Address Register.

**Table 49.22 Address Used by Configuration Set Command for Secure alias (Code flash memory)**

Address	FSADDR Register Value	Setting Data	Operation of additional writing		Timing when the Setting is Enabled
			SAS.FSPR bit is 1	SAS.FSPR bit is 0	
0x0300_A100	0x0300_A100	Option Function Select Register 0 (OFS0)	Writable	Writable	At a reset
0x0300_A104		Option Function Select Register 2 (OFS2)			
0x0300_A110	0x0300_A110	Dual Mode Select Register (DUALSEL)	Writable	Writable	At a reset
0x0300_A134	0x0300_A130	Start-up Area Setting Register (SAS)	Writable	Not writable*1	When a reset or command is executed
0x1300_A180	0x1300_A180	Option Function Select Register 1 (OFS1)	Writable	Writable	At a reset
0x1300_A190	0x1300_A190	Bank Select Register (BANKSEL)	Writable	Writable	At a reset
0x1300_A1C0	0x1300_A1C0	Block Protect Setting Register (BPS)	Writable*2	Writable*2	When a reset or command is executed
0x1300_A1E0	0x1300_A1E0	Permanent Block Protect Setting Register (PBPS)	Writable*3 (from 1 to 0 only)	Writable*3 (from 1 to 0 only)	When a reset or command is executed
0x0300_A200	0x0300_A200	Option Function Select Register 1 Secure (OFS1_SEC)	Writable	Writable	At a reset
0x0300_A210	0x0300_A210	Bank Select Register Secure (BANKSEL_SEC)	Writable	Writable	At a reset
0x0300_A240	0x0300_A240	Block Protect Setting Register Secure (BPS_SEC)	Writable*4	Writable*4	When a reset or command is executed
0x0300_A260	0x0300_A260	Permanent Block Protect Setting Register Secure (PBPS_SEC)	Writable*5 (from 1 to 0 only)	Writable*5 (from 1 to 0 only)	When a reset or command is executed
0x0300_A280	0x0300_A280	Option Function Select Register 1 Select (OFS1_SEL)*6	Writable	Writable	At a reset
0x0300_A290	0x0300_A290	Bank Select Register Select (BANKSEL_SEL)	Writable	Writable	At a reset
0x0300_A2C0	0x0300_A2C0	Block Protect Setting Register Select (BPS_SEL)	Writable*6	Writable*6	At a reset

- Note 1. The SAS.FSPR bit cannot be restored to 1 by using the Configuration set command once it is set to 0. Therefore, setting the start-up area select flags again becomes impossible. (when the Configuration set command is issued to the address of 0x0100A134, the command is locked.) Exercise extra caution when handling the SAS.FSPR bit.
- Note 2. Once PBPS[n] bit is set to 0, the BPS[n] bit cannot be restored to 1 by using the Configuration set command.
- Note 3. Once these bits are set to 0, the bits cannot be restored to 1 by using the Configuration set command. The PBPS[n] bit cannot be set to 0 by using the Configuration set command when the BPS[n] bit is 1.
- Note 4. Once PBPS\_SEC[n] bit is set to 0, the BPS\_SEC[n] bit cannot be restored to 1 by using the Configuration set command.
- Note 5. Once these bits are set to 0, the bits cannot be restored to 1 by using the Configuration set command. The PBPS\_SEC[n] bit cannot be set to 0 by using the Configuration set command when the BPS\_SEC[n] bit is 1.
- Note 6. When the Permanent Block Protect bit is enabled, Block protect setting register cannot be written.

**Table 49.23 Address Used by Configuration Set Command for Secure alias (Data flash memory) (1 of 2)**

Address	FSADDR	Setting Data	Operation of additional writing		Timing when the Setting is Enabled
	Register Value		Lock bit is 1	Lock bit is 0	
0x2703_0098~ 0x2703_017C	0x2703_0098~ 0x2703_017C	Lockable Configuration Data Area 0 (LK_CD_A0)	Writable*1	Not Writable*1	At a reset
0x2703_0180~ 0x2703_027C	0x2703_0180~ 0x2703_027C	Lockable Configuration Data Area 1 (LK_CD_A1)	Writable*1	Not Writable*1	At a reset
0x2703_0280~ 0x2703_035C	0x2703_0280~ 0x2703_035C	Lockable Configuration Data Area 2 (LK_CD_A2)	Writable*1	Not Writable*1	At a reset

**Table 49.23 Address Used by Configuration Set Command for Secure alias (Data flash memory) (2 of 2)**

Address	FSADDR	Setting Data	Operation of additional writing		Timing when the Setting is Enabled
	Register Value		Lock bit is 1	Lock bit is 0	
0x2703_0380~ 0x2703_0384	0x2703_0380~ 0x2703_0384	Configuration Data Lock Bit Area 0 (CD0_LK)	Writable	Writable	When a reset or command is executed
0x2703_0388~ 0x2703_038C	0x2703_0388~ 0x2703_038C	Configuration Data Lock Bit Area 1 (CD1_LK)	Writable (from 1 to 0 only)	Writable (from 1 to 0 only)	When a reset or command is executed
0x2703_0390	0x2703_0390	Configuration Data Lock Bit Area 2 (CD2_LK)	Writable (from 1 to 0 only)	Writable (from 1 to 0 only)	When a reset or command is executed
0x2703_03C0	0x2703_03C0	Anti-Rollback Counter Lock Setting (ARCLS)	Writable (from 1 to 0 only)	Writable (from 1 to 0 only)	When a reset or command is executed
0x2703_03C2		Anti-Rollback Counter Configuration Setting for ARC_NSEC (ARCCS)	Writable*2	Writable*2	When a reset or command is executed

Note 1. For details of relation of write-protection area and lock bit, see [section 49.12.4. Data Flash Configuration Area Protection](#).

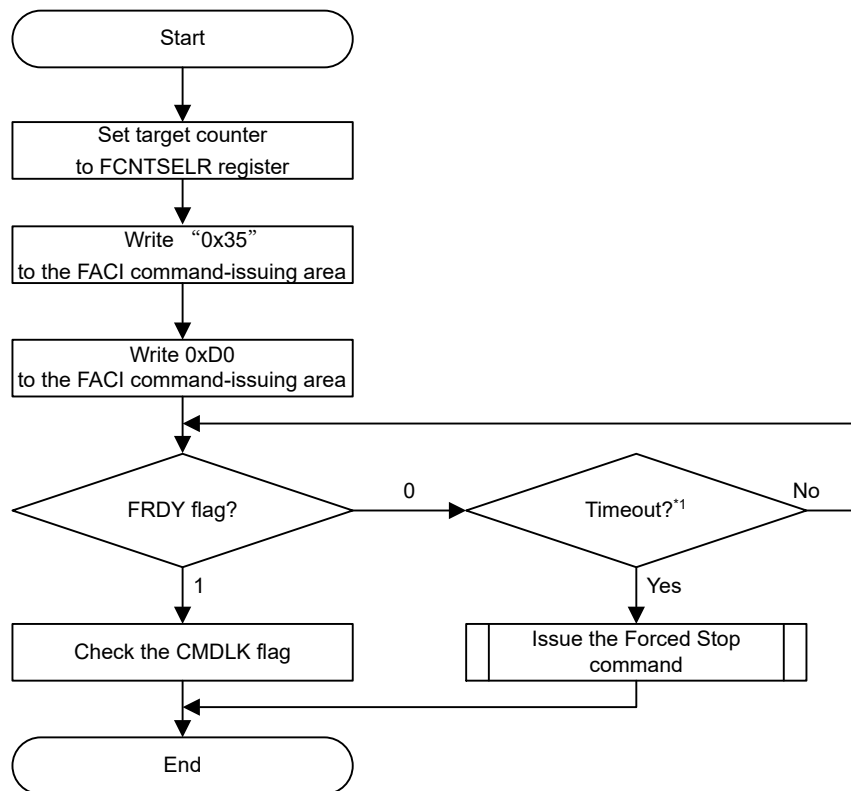
Note 2. The ARC\_NSEC counter selection bit cannot be changed after set to 00b or 01b. When using the ARC\_NSEC counter, this counter can be selected either of 4 lines of 64 bits (00b setting) or 1 line of 256 bits (01b setting).

#### 49.9.3.16 Increment Counter Command

The Increment Counter command is used to security function, safety function. "Increment Counter" command is used for updating 2 types of anti-rollback counter (ARC\_SEC and ARC\_NSEC). The counter update is executed increment processing and cannot be decrement processing.

Writing "0xD0" at 2nd write access of flash sequencer command triggers FACI to start "Increment Counter" command processing. Completion of command processing can be confirmed by FRDY bit of FSTATR register.

If "Command Lock" status occurs during command execution, refresh the counter with "Refresh Counter" command after "Command Lock" state is released.



Note 1. 50msec(FCLK $\geq$ 20MHz)

**Figure 49.26 Usage of the Increment Counter Command**

Anti-rollback counter value is not increased the following conditions by "Increment Counter" command.

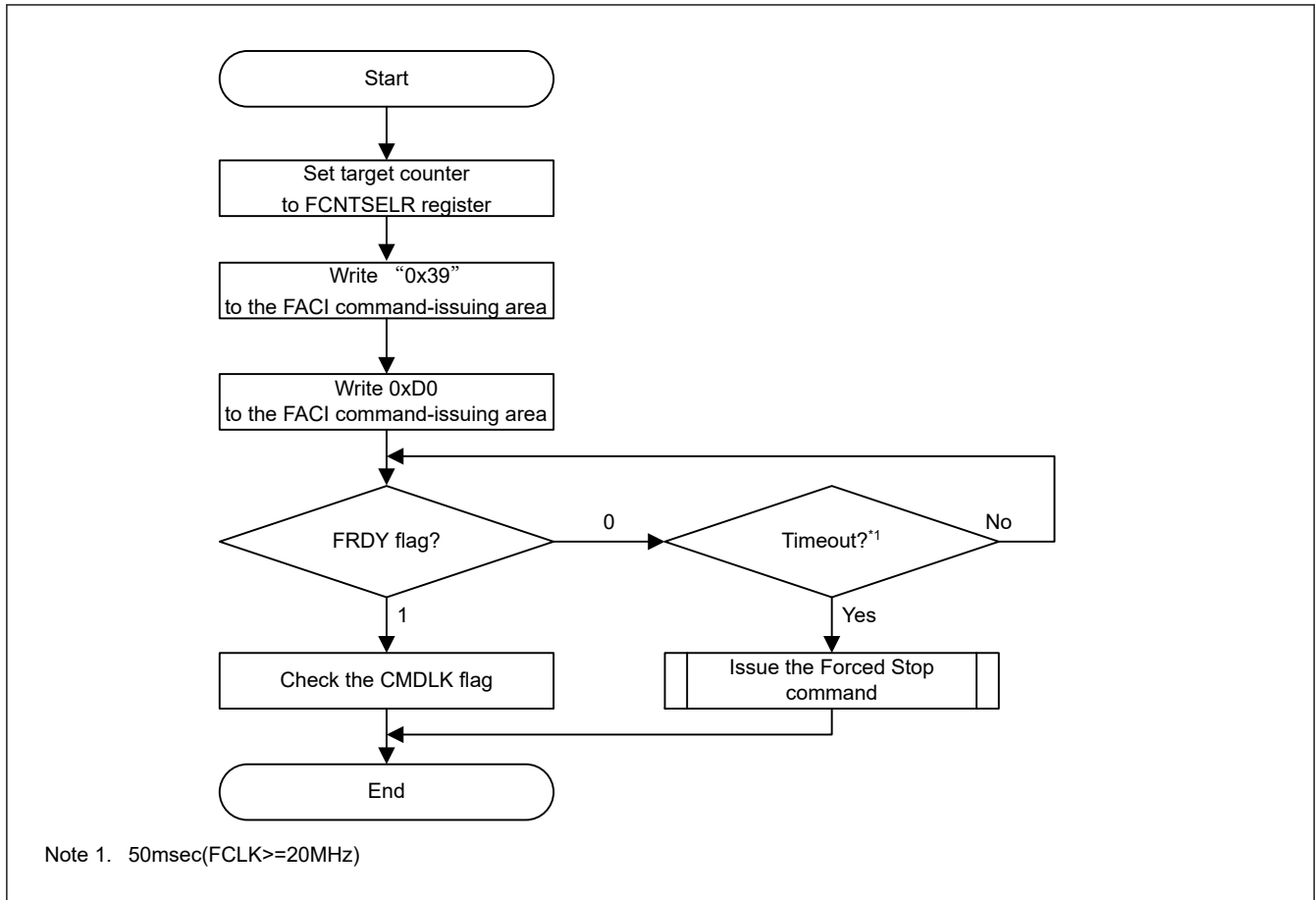
- Target counter is protected by ARCSEC\_LK or ARCNSEC\_LK0~3.
- Target counter value is maximum. (Error does not occur)
- Target counter is ARC\_NSEC when CNF\_ARCNSEC0~1 is invalid setting.

#### 49.9.3.17 Refresh Counter Command

The Refresh Counter command is used to security function, safety function. "Refresh Counter" command is used for update of anti-rollback counter when power failure is occurred while "Increment Counter" command processing. Counter value is not changed by "Refresh Counter" command.

Writing "0xD0" at 2nd write access of flash sequencer command triggers FACL to start "Increment Counter" command processing. Completion of command processing can be confirmed by FRDY bit of FSTATR register.

If "Command Lock" status occurs during command execution, refresh the counter with "Refresh Counter" command after "Command Lock" state is released.

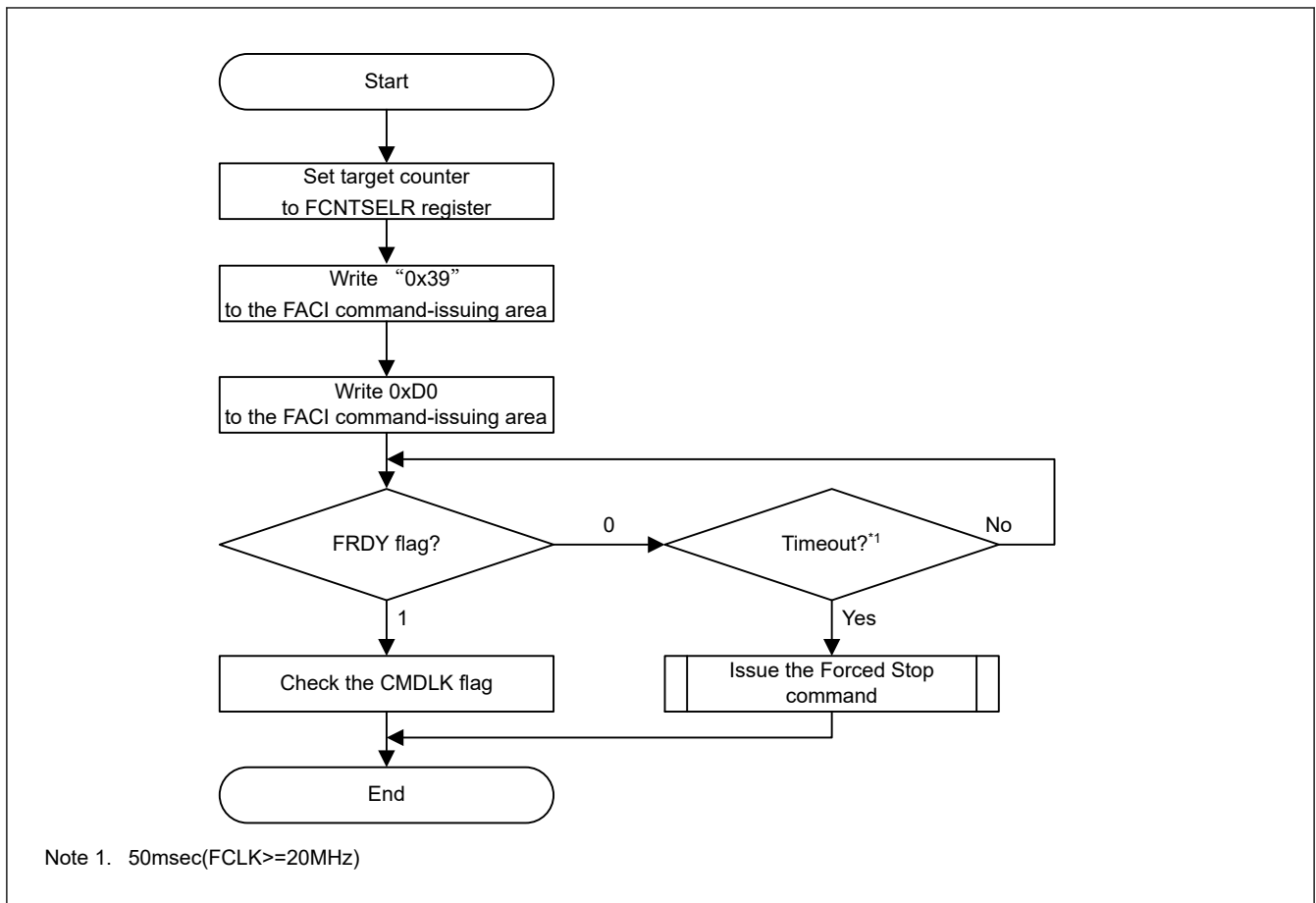


**Figure 49.27 Usage of the Refresh Counter Command**

#### 49.9.3.18 Read Counter Command

The Read Counter command is used to security function, safety function. "Read Counter" command is used for reading 2 types of anti-rollback counter (ARC\_SEC and ARC\_NSEC). Flash sequencer automatically judges valid area of anti-rollback counter, reads 64 bits data, and stores to FCNTDATAR0 and FCNTDATAR1 registers.

Writing "0xD0" at 2nd write access of flash sequencer command triggers FACL to start "Read Counter" command processing. Completion of command processing can be confirmed by FRDY bit of FSTATR register.



**Figure 49.28 Usage of the Read Counter Command**

## 49.10 Suspend Operation

Reading from the flash memory is not possible during programming or erasure if the conditions for background operation given in [Table 49.39](#) are not satisfied. When a P/E suspend command is issued to suspend the programming or erasure of the flash memory, reading from the flash memory is enabled. Regarding P/E suspend commands, there are one suspend command mode for programming and two suspend command modes for erasure (suspension priority mode and erasure priority mode). To resume suspended programming or erasure, the P/E resume command is available. For details on the suspend operation, see [Figure 49.17](#).

## 49.11 Protection Function

### 49.11.1 Software Protection

Software protection disables programming and erasure of the code flash memory through the settings of control registers and block protect setting in the user area. If an attempt is made to issue an FACL command against software protection, the flash sequencer enters the command-locked state.

#### 49.11.1.1 Protection through FWEPROR

Unless the FWEPROR.FLWE[1:0] bits are set to 01b, programming cannot proceed in any mode.

#### 49.11.1.2 Protection by FENTRYR

When the FENTRYR register is set to 0x0000, the flash sequencer enters read mode. In read mode, FACL commands cannot be accepted. If an attempt is made to issue an FACL command in read mode, the flash sequencer enters the command-locked state.



### 49.11.1.3 Protection by Block Protect Setting

Each block in user area has the block protect setting (BPS or BPS\_SEC). When the FBPROT0 or FBPROT1 register is 0x0000 and the block protect bit is 0, issuing the Program or Block Erase command to user area of the code flash causes the command-locked state. To program or erase the block whose block protect bit is 0, set the FBPROT0 or FBPROT1 register to 0x0001.

The block protect setting can be locked by the permanent block protect setting (PBPS or PBPS\_SEC). When the permanent block protect setting and the block protect setting are 0, issuing a Program or Block erase command to user area of the code flash causes the flash sequencer to enter the command-locked state regardless of the FBPROT0 and FBPROT1 register settings.

Valid block protect setting (BPS or BPS\_SEC) depends on the Block Protect Select bit (BPS\_SEL).

See [section 49.12.2. Permanent Block Protect Setting](#) for details of the block protect setting and permanent block protect setting. See [section 49.4.17. FBPROT0 : Flash Block Protection Register](#) and [section 49.4.18. FBPROT1 : Flash Block Protection for Secure Register](#) for more information.

For details of block protect setting (BPS or BPS\_SEC) and block protect select (BPS\_SEL), see [section 6, Option-Setting Memory](#).

The protected area by the block protect setting is always determined by the address of the FSADDR register setting regardless of the address swapping function setting (the startup area select, the block swap in linear mode, and the startup bank select in dual mode). [Table 49.24](#) to [Table 49.29](#) show the relation of user area and the block protect setting in each function setting.

#### (1) In case of the linear mode

- BPS[0] to BPS[n] or BPS\_SEC[0] to BPS\_SEC[n] are assigned to the block of user area (for example, address is 0x00\_0000 to the last block address).
- BPS[0]/BPS\_SEC[0] and BPS[1]/BPS\_SEC[1] are assigned to the block of user area depending on the startup area select setting (SAS.BTFLG bit). (See [section 49.11.3. Start-Up Program Protection](#)).
- The second half of FLI user area is assigned to block protect setting depending on the block swap select setting (BANKSEL.BLCKSWP[3:0] bits). See [section 49.11.5. Block Swap Function](#).

[Table 49.24](#) shows the block protect setting when the startup area select is disabled (not swapping).

[Table 49.25](#) and later show example of the block protect setting when the address conversion function is used.

**Table 49.24 Example of Block Protect setting when SAS.BTFLG is 1**

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address	32 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 9	—
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 8	—
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 7	—
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 6	—
⋮	⋮	⋮	⋮	—
0x00_2000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 1	Not swap block 0 and block 1 in this startup area select setting
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 0	Not swap block 0 and block 1 in this startup area select setting

**Table 49.25 Example of Block Protect setting when SAS.BTFLG is 0**

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address	32 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 9	—
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 8	—
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 7	—
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 6	—
⋮	⋮	⋮	⋮	—
0x00_2000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 0	Swap block 0 and block 1 in this startup area select setting
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 1	Swap block 0 and block 1 in this startup area select setting

**Table 49.26 Example of Block Protect setting when BANKSEL.BLCKSWP is 1**

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address	32 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
The upper side address of the block swap target	32 KB	BPS[b] or BPS_SEC[b]	Block b	Not swap block a and block b in this block swap select setting
⋮	⋮	⋮	⋮	—
The lower side address of the block swap target	32 KB	BPS[a] or BPS_SEC[a]	Block a	Not swap block a and block b in this block swap select setting
⋮	⋮	⋮	⋮	—
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 9	—
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 8	—
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 7	—
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 6	—
⋮	⋮	⋮	⋮	—
0x002000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 1	—
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 0	—

**Table 49.27 Example of Block Protect setting when BANKSEL.BLCKSWP is 0 (1 of 2)**

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address	32 KB	BPS[n] or BPS_SEC[n]	Block n	—
⋮	⋮	⋮	⋮	—
The updating side address of the block swap target	32 KB	BPS[b] or BPS_SEC[b]	Block a	Swap block a and block b in this block swap select setting
⋮	⋮	⋮	⋮	—
The operating side address of the block swap target	32 KB	BPS[a] or BPS_SEC[a]	Block b	Swap block a and block b in this block swap select setting

**Table 49.27 Example of Block Protect setting when BANKSEL.BLCKSWP is 0 (2 of 2)**

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
⋮	⋮	⋮	⋮	—
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 9	—
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 8	—
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 7	—
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 6	—
⋮	⋮	⋮	⋮	—
0x00_2000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 1	—
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 0	—

## (2) In case of the dual mode (DUALSEL.BANKMD[2:0] = 000b)

- BPS[0] to BPS[n] or BPS\_SEC[0] to BPS\_SEC[n] are assigned the block of bank 0 user area (for example, address is 0x00000000 to the last block address in the lower side bank).
- BPS[70] to BPS[70 + n] or BPS\_SEC[70] to BPS\_SEC[70 + n] are assigned to the block of bank 1 user area (for example, address is 0x00200000 to the last block address in the upper side bank).

Bank0 is upper side bank (BANKSEL.BANKSWP[2:0] = 000b)

- BPS[0] to BPS[n] or BPS\_SEC[0] to BPS\_SEC[n] are assigned to the block of bank1 user area (e.g. Address is 0x00\_0000 to The last block address in lower side bank).
- BPS[70] to BPS[70+n] or BPS\_SEC[70] to BPS\_SEC[70+n] are assigned to the block of bank0 user area (e.g. Address is 0x20\_0000 to The last block address in upper side bank).

Table 49.28 and Table 49.29 show example of the block protect setting in the dual mode. See section 49.11.4. Dual Bank Function for details of dual bank function (DUALSEL.BANKMD[2:0] and BANKSEL.BANKSWP[2:0] bits).

**Table 49.28 Example of Block Protect setting when BANKSEL.BANKSWP[2:0] is 111b (1 of 2)**

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address in upper side bank	32 KB	BPS[70 + n] or BPS_SEC[70 + n]	Block 70 + n	Not swap bank 0 and bank 1 in this startup bank switch setting
⋮	⋮	⋮	⋮	
0x21_8000 to 0x21_FFFF	32 KB	BPS[79] or BPS_SEC[79]	Block 79	
0x21_0000 to 0x21_7FFF	32 KB	BPS[78] or BPS_SEC[78]	Block 78	
0x20_E000 to 0x20_FFFF	8 KB	BPS[77] or BPS_SEC[77]	Block 77	
0x20_C000 to 0x20_DFFF	8 KB	BPS[76] or BPS_SEC[76]	Block 76	
⋮	⋮	⋮	⋮	
0x20_2000 to 0x20_3FFF	8 KB	BPS[71] or BPS_SEC[71]	Block 71	
0x20_0000 to 0x20_1FFF	8 KB	BPS[70] or BPS_SEC[70]	Block 70	

**Table 49.28 Example of Block Protect setting when BANKSEL.BANKSWP[2:0] is 111b (2 of 2)**

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address in lower side bank	32 KB	BPS[n] or BPS_SEC[n]	Block n	Not swap bank 0 and bank 1 in this startup bank switch setting
⋮	⋮	⋮	⋮	
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 9	
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 8	
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 7	
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 6	
⋮	⋮	⋮	⋮	
0x00_2000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 1	
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 0	

**Table 49.29 Relation of User area and Block Protect setting when BANKSEL.BANKSWP[2:0] is 000b**

FSADDR[23:0]	Block size	Block protect setting	User area block number	Notes
The last block address in the upper side bank	32 KB	BPS[70 + n] or BPS_SEC[70 + n]	Block n	Swap bank 0 and bank 1 in this startup bank switch setting
⋮	⋮	⋮	⋮	
0x21_8000 to 0x21_FFFF	32 KB	BPS[79] or BPS_SEC[79]	Block 9	
0x21_0000 to 0x21_7FFF	32 KB	BPS[78] or BPS_SEC[78]	Block 8	
0x20_E000 to 0x20_FFFF	8 KB	BPS[77] or BPS_SEC[77]	Block 7	
0x20_C000 to 0x20_DFFF	8 KB	BPS[76] or BPS_SEC[76]	Block 6	
⋮	⋮	⋮	⋮	
0x20_2000 to 0x20_3FFF	8 KB	BPS[71] or BPS_SEC[71]	Block 1	
0x20_0000 to 0x20_1FFF	8 KB	BPS[70] or BPS_SEC[70]	Block 0	
The last block address in the lower side bank	32 KB	BPS[n] or BPS_SEC[n]	Block 70 + n	Swap bank 0 and bank 1 in this startup bank switch setting
⋮	⋮	⋮	⋮	
0x01_8000 to 0x01_FFFF	32 KB	BPS[9] or BPS_SEC[9]	Block 79	
0x01_0000 to 0x01_7FFF	32 KB	BPS[8] or BPS_SEC[8]	Block 78	
0x00_E000 to 0x00_FFFF	8 KB	BPS[7] or BPS_SEC[7]	Block 77	
0x00_C000 to 0x00_DFFF	8 KB	BPS[6] or BPS_SEC[6]	Block 76	
⋮	⋮	⋮	⋮	
0x00_2000 to 0x00_3FFF	8 KB	BPS[1] or BPS_SEC[1]	Block 71	
0x00_0000 to 0x00_1FFF	8 KB	BPS[0] or BPS_SEC[0]	Block 70	

### 49.11.2 Error Protection

Error protection detects the issuing of illegal FACI commands, illegal access, and flash sequencer malfunction. FACI command acceptance is disabled (command-locked state) in response to the detection of these errors. The flash memory cannot be programmed or erased while the flash sequencer is in the command-locked state. For release from the command-locked state, issue the Status Clear or Forced Stop command. The Status Clear command can only be used while the FRDY bit in the FSTATR register is 1. The Forced Stop command can be used regardless of the value of the FRDY bit. While the CMDLKIE bit in the FAEINT register is 1, a flash access error (FIFERR) interrupt is generated if the flash sequencer enters the command-locked state (the CMDLK bit in the FFASTAT register is set to 1).

If the flash sequencer enters the command-locked state in response to a command other than the P/E suspend command during programming or erasure processing, the flash sequencer continues the processing for programming or erasure. In this

state, the P/E suspend command cannot be used to suspend the processing for programming or erasure. If a command is issued in the command-locked state, the ILGLERR bit becomes 1 and the other bits retain the values set from previous error detection.

Table 49.30 shows the error protection types and status bit values after error detections.

**Table 49.30 Error protection type (1 of 3)**

Error type	Description	ILGOMERR	FESETERR	SECERR	OTERR	TZFERR	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE	
FENTRYR setting error	The value set in FENTRYR is not 0x0000, 0x0001, or 0x0080	0	1	0	0	0	1	0	0	0	0	0	
	The FENTRYR setting at suspension is different from that at resumption	0	1	0	0	0	1	0	0	0	0	0	
Illegal command error	An undefined size is specified in the first cycle of the command. (not byte-write)	1	0	0	0	0	1	0	0	0	0	0	
	An undefined code is written in the first access of the FACL command	1	0	0	0	0	1	0	0	0	0	0	
	The value specified in the last access of the multiple-access FACL command is not 0xD0	1	0	0	0	0	1	0	0	0	0	0	
	The value (N) specified in the second write access of the FACL command in the program or configuration set command is wrong	1	0	0	0	0	1	0	0	0	0	0	
	Blank Check command is issued with inconsistent BCDIR, FSADDR, and FEADDR settings (see <a href="#">section 49.4.13. FEADDR : FACL Command End Address Register</a> )	1	0	0	0	0	1	0	0	0	0	0/1*1	
	A multi block erase command is issued with inconsistent FSADDR and FEADDR settings. <ul style="list-style-type: none"> <li>FSADDR &gt; FEADDR</li> <li>FEADDR is set to reserved area.</li> </ul>	1	0	0	0	0	1	0	0	0	0	0/1*1	
	Increment Counter, Refresh Counter, or Read Counter command is issued with inconsistent FCNTSEL[2:0] setting. (Include CNF_ARCNS0,1 setting is the mismatch.)	1	0	0	0	0	1	0	0	0	0	0	0
	An FACL command not acceptable in each mode is issued (see <a href="#">Table 49.19</a> )	1	0	0	0	0	1	0	0	0	0	0	0
	An FACL command is issued when command acceptance conditions are not satisfied (see <a href="#">Table 49.20</a> )	0/1	0/1	0/1	0/1	0	1	0/1	0/1	0/1	0/1	0/1	0/1
A program, block erase or multi block erase command is issued against the area protected by the block protect setting (see <a href="#">section 49.11.1.3. Protection by Block Protect Setting</a> )	1	0	0	0	0	1	0	0	0	0	0	0	

Table 49.30 Error protection type (2 of 3)

Error type	Description	ILGOMERR	FESETERR	SECERR	OTERR	TZFERR	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE
Illegal command error	Configuration set command is issued against Data flash configuration area protected by lock bit setting	1	0	0	0	0	1	0	0	0	0	0
	Increment Counter or Refresh Counter command is issued against anti-rollback area protected by lock bit setting	1	0	0	0	0	1	0	0	0	0	0
	A program command is issued against the erase area in erase suspend	1	0	0	0	0	1	0	0	0	0	0
	A command is issued against flash sequencer command illegal status(see <a href="#">Table 49.17</a> ).	0/1	0/1	0/1	0/1	0	1	0/1	0/1	0/1	0/1	0/1
Erase error	An error occurs during erasure	0	0	0	0	0	0	1	0	0	0	0
Programming error	An error occurs during programming	0	0	0	0	0	0	0	1	0	0	0
Code flash memory access violation	An FACL command is issued to the reserved portion of the user area in code flash P/E mode	0	0	0	0	0	1	0	0	0	1	0
	Configuration set command is issued to the reserved configuration area in code flash P/E mode.	0/1	0	0	0	0	1	0	0	0	1	0
	Program, block erase, or configuration set command is issued to TrustZone protection area in code flash P/E mode.	0/1	0	0	0	1	1	0	0	0	1	0
Data flash memory access violation	A program or block erase command is issued to the reserved data area in data flash P/E mode	0	0	0	0	0	1	0	0	0	0	1
	A multi block erase command is issued to the reserved data area in data flash P/E mode. (FSADDR is set to reserved data area).	1	0	0	0	0	1	0	0	0	0	1
	Blank Check command is issued to reserved data area in data flash P/E mode. (FSADDR is set to reserved data area ).	1	0	0	0	0	1	0	0	0	0	1
	Configuration set command has been issued to reserved configuration area in data flash P/E mode.	0/1	0	0	0	0	1	0	0	0	0	1
	A program, block erase, multi block erase, blank check, configuration set, increment counter, refresh counter, or read counter command is issued to TrustZone protection area in data flash P/E mode.	0	0	0	0	0	1	0	0	0	0	1
Security error	Configuration set command for the SAS.BTFLG bit setting is issued when the SAS.FSPR bit is 0 (see <a href="#">section 49.9.3.15. Configuration Set Command</a> )	0	0	1	0	0	1	0	0	0	0	0

**Table 49.30 Error protection type (3 of 3)**

Error type	Description	ILGOMERR	FESETERR	SECERR	OTERR	TZFERR	ILGLERR	ERSERR	PRGERR	FLWEERR	CFAE	DFAE
Others	An FACI command-issuing area is accessed in read mode	0	0	0	1	0	1	0	0	0	0	0
	An FACI command-issuing area is read in code flash P/E mode or data flash P/E mode	0	0	0	1	0	1	0	0	0	0	0
Flash write erase protection error	A flash memory write protection error is detected by the FWEPROR register setting <sup>2</sup> during command processing by the flash sequencer	0	0	0	0	0	0	0/1	0/1	1	0	0

Note 1. DFAE value depends on the FSADDR setting.

Note 2. For details on the FWEPROR register, see [section 49.4.8. FWEPROR : Flash P/E Protect Register](#).

### 49.11.3 Start-Up Program Protection

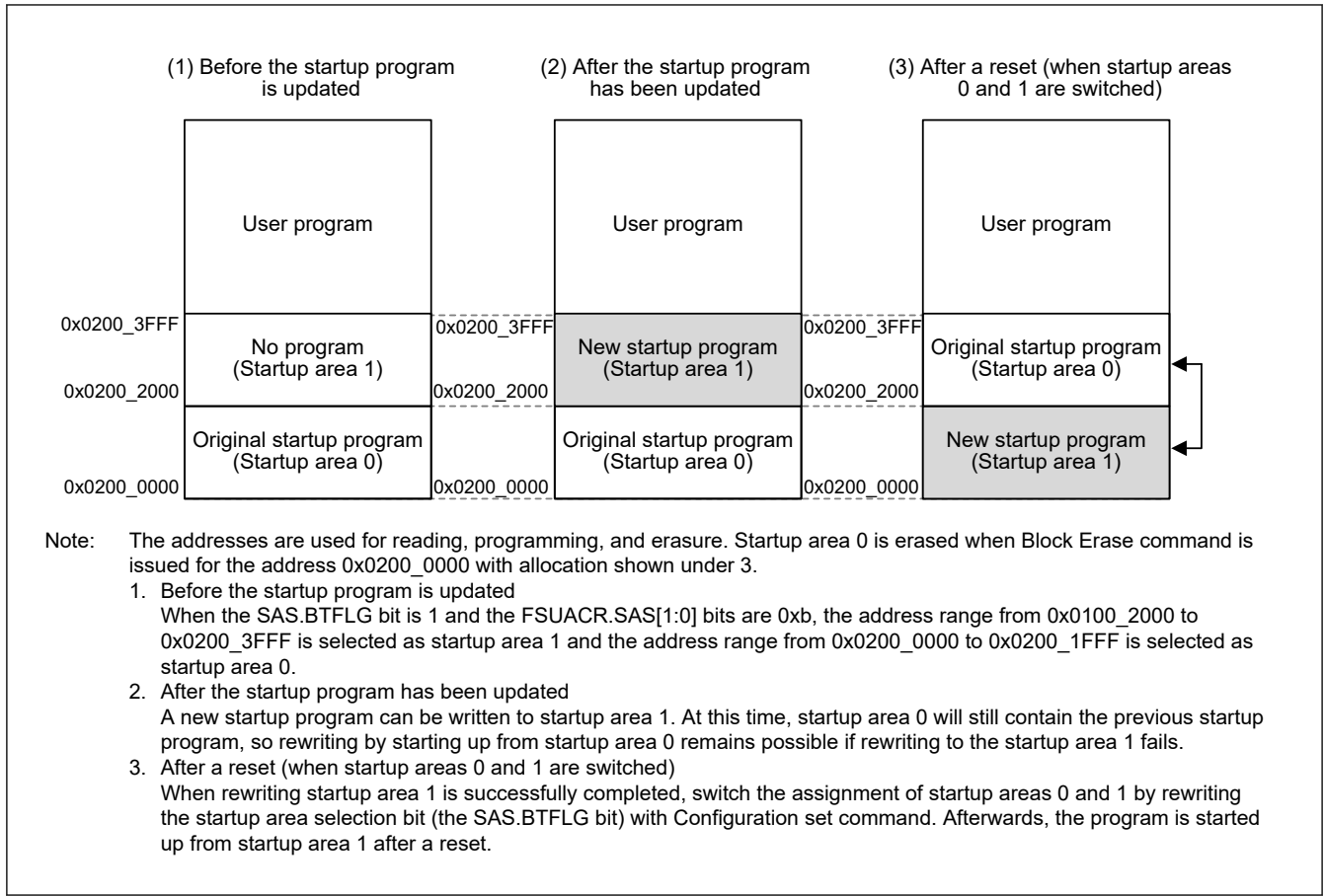
Protection of the startup program is for protection of the program to be started after a reset (the startup program). This function provides a way to safely update the startup program when rewriting is suspended during a reset.

The startup area is 8 Kbytes in size and is assigned to the user area in the code flash memory. This function uses the values of the SAS.BTFLG bit and the FSUACR.SAS[1:0] bits to change the area where the startup program is stored in block units (see [Figure 49.29](#) to [Figure 49.32](#)).

In protection of the startup program, the state of the selection of the startup area can be fixed by the FSPR bit. However, the SAS.FSPR bit never be restored to 1 once the flag is set to 0. Exercise extra caution when handling the SAS.FSPR bit.

In addition, this protection cannot be used when dual mode is selected by the bank mode switching function (when the BANKMD[2:0] bits are 000b).

For details of SAS.FSPR bit and DUALSEL.BANKMD[2:0] bits, see [section 6, Option-Setting Memory](#).



**Figure 49.29 Concept of Protection of the Startup Program for Secure alias**



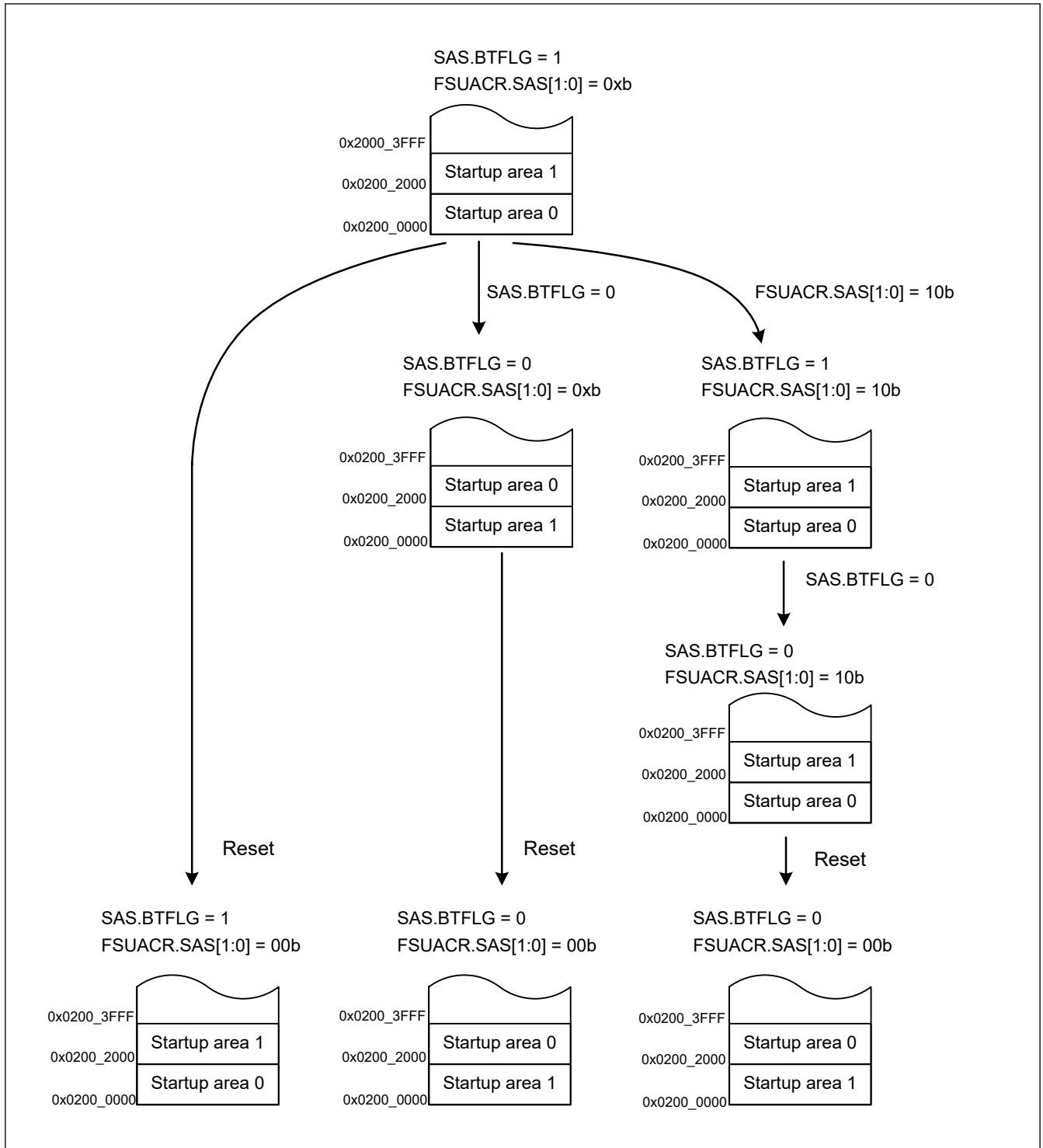


Figure 49.30 Example 1 of Transitions for Startup Program Protection Settings for Secure alias

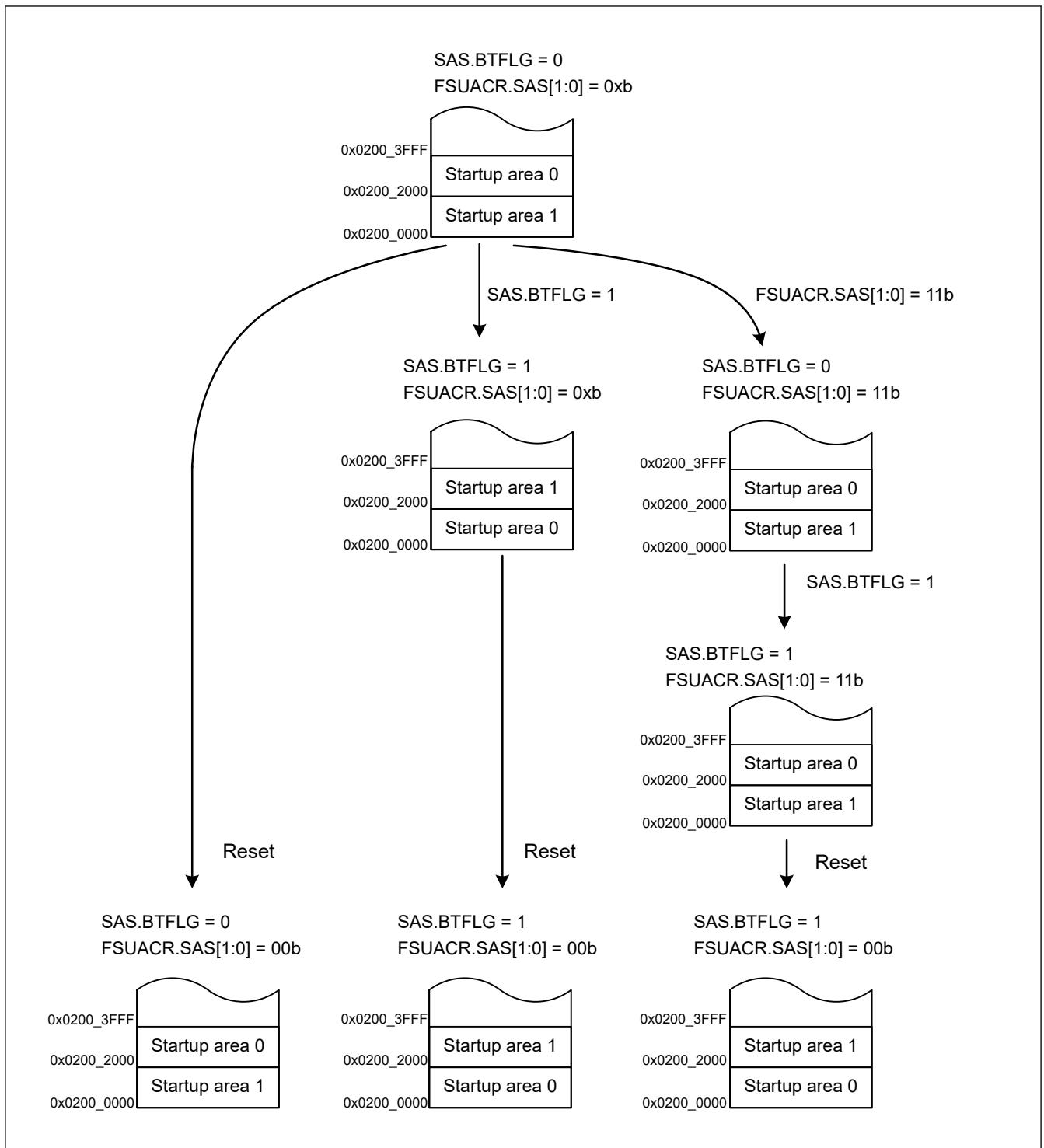
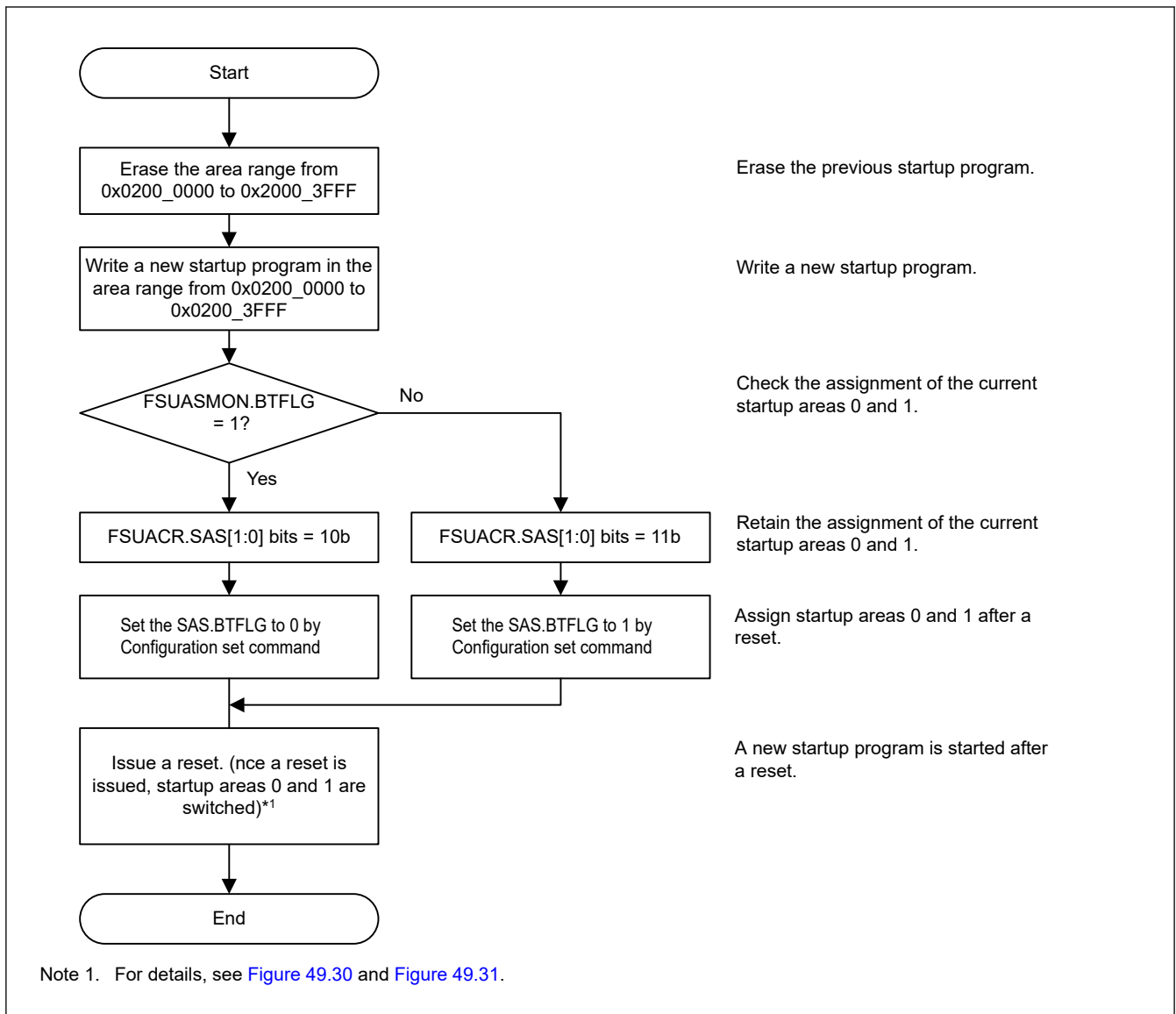


Figure 49.31 Example 2 of Transitions for Startup Program Protection Settings for Secure alias



**Figure 49.32 Concept of Protection of the Startup Program for Secure alias**

#### 49.11.4 Dual Bank Function

This protection uses the functions of bank mode switching and startup bank selection to update a program while a user program is running and to provide a safe method of updating in cases where programming is suspended during a reset.

##### 49.11.4.1 Switching Bank Modes

The bank mode switching function selects either linear mode in which the user area in the code flash memory is used as one area, or dual mode in which the user area is divided into two bank areas. [Figure 49.33](#) shows an example of flow of switching bank modes. A reset after setting the DUALSEL.BANKMD[2:0] bits in the option setting memory determines the mode of the bank mode switching function. Selecting dual mode enables the startup bank selection function.

When dual mode is selected by bank mode switching function (the DUALSEL.BANKMD[2:0] bits are 000b), start-up program protection function cannot be used.

For details of DUALSEL.BANKMD[2:0] bits, see [section 6, Option-Setting Memory](#).

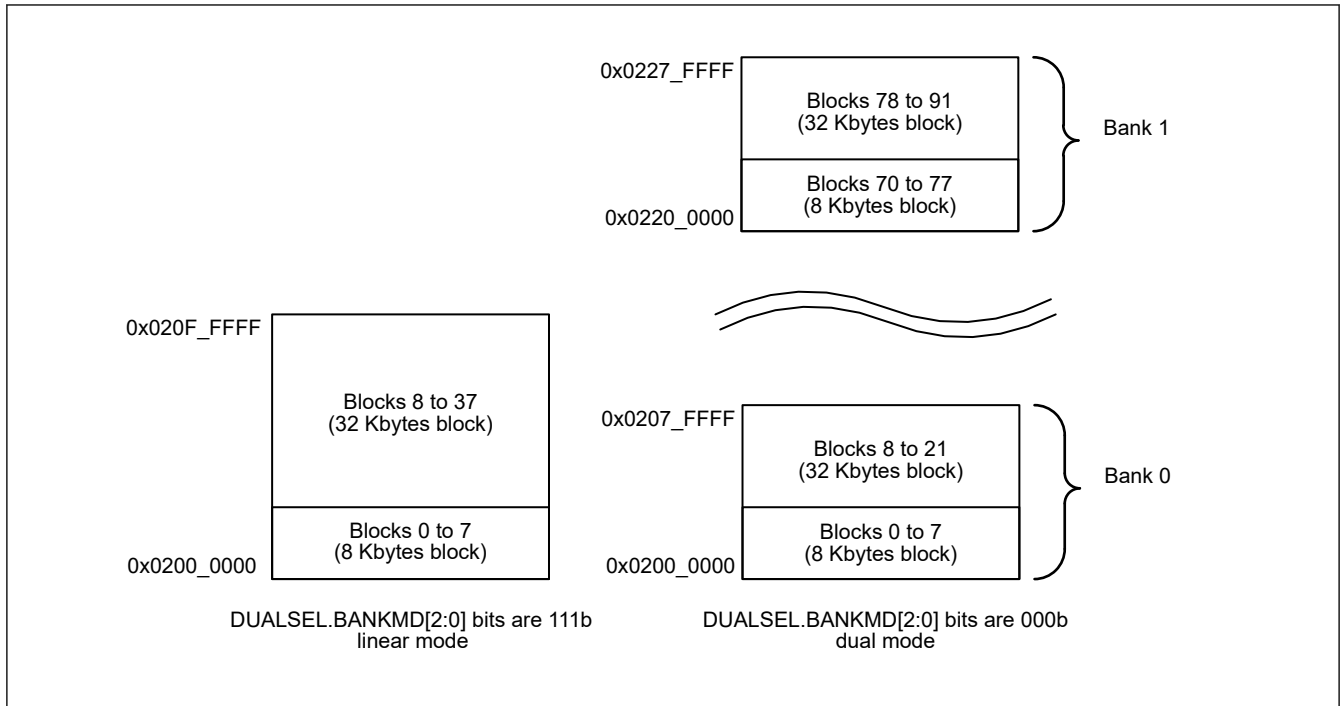


Figure 49.33 Example of Flow of Switching Bank Modes for Secure alias

#### 49.11.4.2 Selecting the Startup Bank

Startup bank selection provides a way to safely update the program by selecting a bank area to be started in dual mode (when the DUALSEL.BANKMD[2:0] bits are 000b) when programming is suspended during a reset. Figure 49.34 is a schematic view of startup bank selection and Table 49.31 shows an example of the flow of startup bank selection. A reset after setting the value of the BANKSEL.BANKSWP[2:0] bits in the option-setting memory changes the addresses of banks 0 and 1 and booting up a program proceeds from the updated area. When the address is switched by using startup bank selection, the programming/erasure target for the FACI commands is also switched. This function is invalid in linear mode. For details of DUALSEL.BANKMD[2:0] bits and BANKSEL.BANKSWP[2:0] bits, see section 6, Option-Setting Memory.

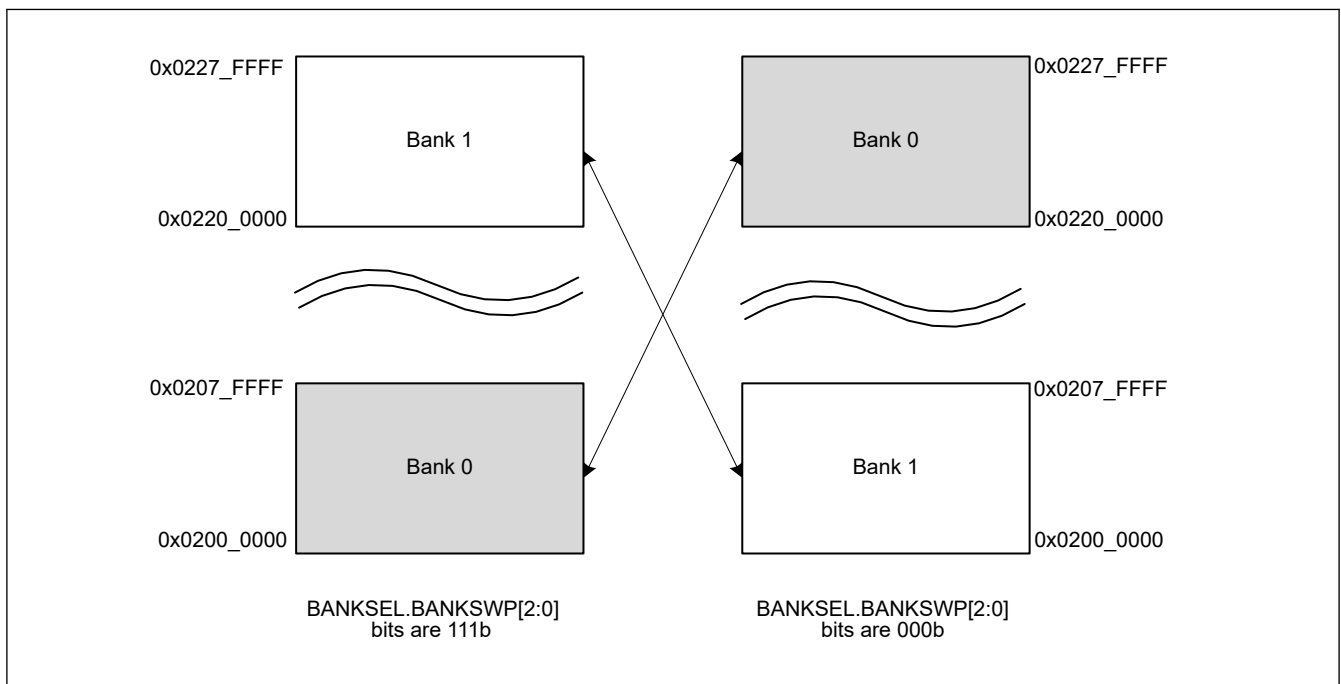


Figure 49.34 Example of Startup Bank Selection for Secure alias

**Table 49.31 Example of Startup Bank Selection Flow for Secure alias**

No.	Step Name	Description
1	Erase block	Erase blocks to be programmed in the address range from 0x0220_0000 to 0x0227_FFFF
2	Program a new software	Program a new software in the address range from 0x0220_0000 to 0x0227_FFFF
3	Read the value	Read the value of the BANKSEL.BANKSWP[2:0] bits.
4	Write an inverted value	Write an inverted value in the BANKSEL.BANKSWP[2:0] bits*1.
5	Issue a reset	Issue a reset (A reset switches the banks.)

Note 1. Set the inverse of the read value of the BANKSEL.BANKSWP[2:0] bits (000b or 111b).

### 49.11.5 Block Swap Function

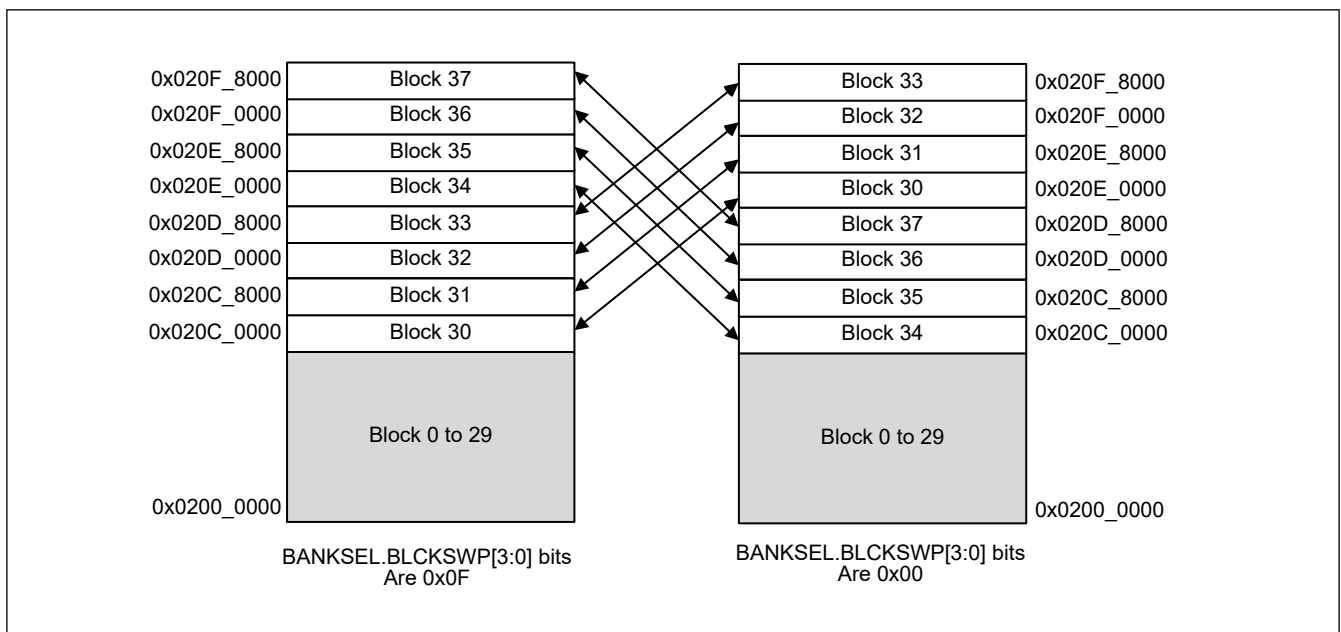
This protection uses the functions of block swap selection to update a program while a user program is running and to provide a safe method of updating in cases where programming is suspended during a reset.

#### 49.11.5.1 Selecting the Block Swap

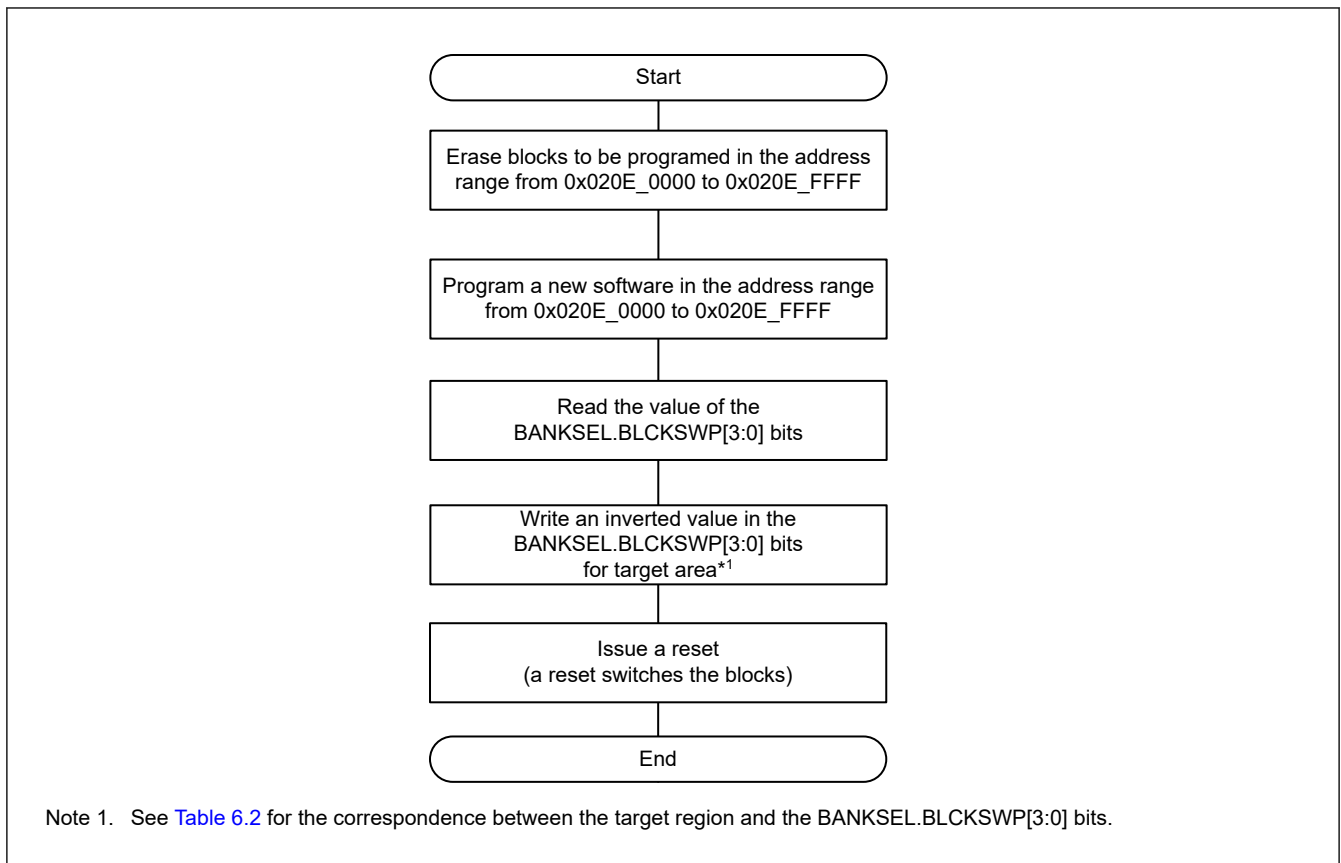
Block swap selection provides a way to safely update the program when programming is suspended during a reset. [Figure 49.35](#) is a schematic view of block swap selection and [Figure 49.36](#) shows an example of the flow of block swap selection. A reset after setting the value of the BANKSEL.BLCKSWP bit in the option-setting memory changes the addresses of block A and B. When the address is switched, the Programming/erasure target for the FACI commands is also switched.

In addition, this protection cannot be used when dual mode is selected by the bank mode switching function (when the DUALSEL.BANKMD[2:0] bits are 000b).

For details of DUALSEL.BANKMD[2:0] bits and BANKSEL.BLCKSWP bit, see [section 6, Option-Setting Memory](#).



**Figure 49.35 Example of Block Swap Selection for Secure alias**



**Figure 49.36 Example of Block Swap Selection Flow for Secure alias**

## 49.12 Security Function

The flash sequencer supports the following security functions:

- Security flag for startup area
- Permanent block protect setting
- Flash memory protection for TrustZone
- Code Flash P/E mode Entry Protection
- Data flash configuration area protection
- Anti-rollback Counter

### 49.12.1 Security Flag for Startup Area Select

The security flag (SAS.FSPR) for the startup area is located in the option-setting memory.

When the SAS.FSPR bit is 0, issuing the configuration set command to change the SAS.BTFLG bit causes the flash sequencer to be in the command-locked state. Also, when the SAS.FSPR bit is 0, it is invalid to write to the Startup Area Select bits SAS[1:0] in the FSUACR register. The SAS.FSPR bit enables protection.

### 49.12.2 Permanent Block Protect Setting

The permanent block protect setting is the clear protection for the block protection setting. User area cannot be permanently updated by the FACL command when the permanent block protect setting is enabled. See [section 49.11.1.3. Protection by Block Protect Setting](#) for more details.

The block protect setting and the permanent block protect setting have the write/clear protection against the configuration set command. The flash sequencer does not detect an error when the configuration set command is issued to the write/clear protected settings.

Figure 49.37 and Table 49.32 show the write/clear protection against the block protect setting (BPS[n]) and the permanent protect setting (PBPS[n]). Figure 49.38 and Table 49.33 show the write/clear protection against the block protect setting for Secure (BPS\_SEC[n]) and permanent protect setting for Secure (PBPS\_SEC[n]).

Effective permanent block protect setting (PBPS or PBPS\_SEC) depends on block protect select (BPS\_SEL). For details of permanent block protect setting (PBPS or PBPS\_SEC) and block protect select (BPS\_SEL), see section 6, Option-Setting Memory.

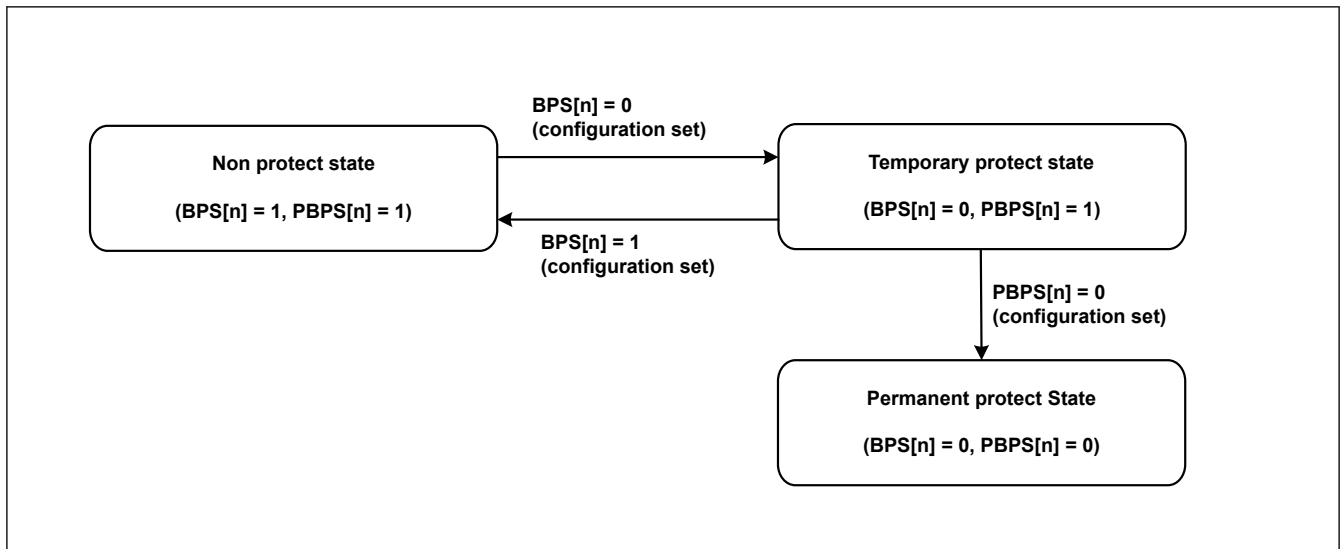


Figure 49.37 Status transition of flash sequencer by BPS[n] and PBPS[n]

Table 49.32 Write/clear protection of BPS[n] , PBPS[n] and BPS\_SEL[n]

Current state		Updatable state by configuration set command				
BPS[n]	PBPS[n]	BPS[n] = 1	BPS[n] = 0	PBPS[n] = 1	PBPS[n] = 0	BPS_SEL[n] = 1
1	1	✓	✓	✓	X	✓
1	0	—	—	—	—	—
0	1	✓	✓	✓	✓	✓
0	0	X	✓	X	✓	X

- Note:
- ✓ indicates updatable by configuration set command.
  - X indicates not updatable by configuration set command (error does not occur).
  - — indicates not reaching to this state.

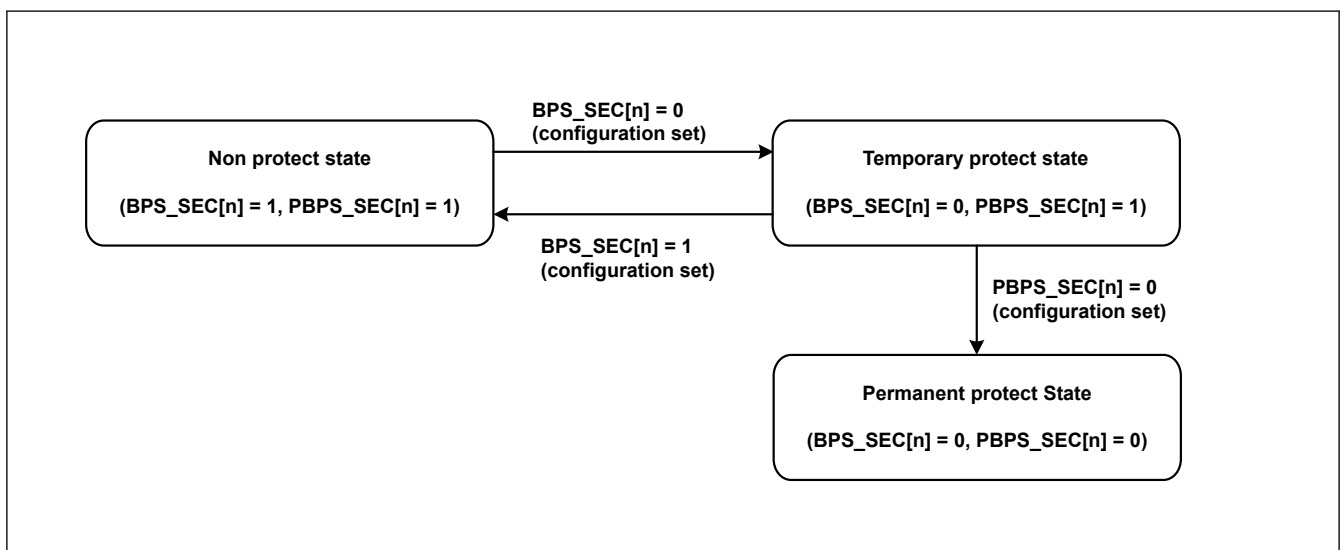


Figure 49.38 Status transition of flash sequencer by BPS\_SEC[n] and PBPS\_SEC[n]

**Table 49.33 Write/clear protection of BPS\_SEC[n], PBPS\_SEC[n] and BPS\_SEL[n]**

Current state		Updatable state by configuration set command				
BPS_SEC[n]	PBPS_SEC[n]	BPS_SEC[n] = 1	BPS_SEC[n] = 0	PBPS_SEC[n] = 1	PBPS_SEC[n] = 0	BPS_SEL[n] = 0
1	1	✓	✓	✓	X	✓
1	0	—	—	—	—	—
0	1	✓	✓	✓	✓	✓
0	0	X	✓	X	✓	X

Note:

- ✓ indicates updatable by configuration set command.
- X indicates not updatable by configuration set command (error does not occur).
- — indicates not reaching to this state.

### 49.12.3 Flash Memory Protection for TrustZone

Information in this section focuses on the flash sequencer operation.

The flash memory provides the following types of protect function against Non-secure access:

- Protection for flash memory area (P/E)
- Protection for flash memory area (read)
- Protection for registers
- Code flash P/E mode entry protection

#### 49.12.3.1 Protection for Flash Memory Area (P/E)

This function protects the Secure region of the code flash and data flash from FACI commands of Non-secure access. The condition of protection depends on the FACI command, the access attribution, and the memory boundary setting.

For details of Secure region, see [section 40, Security Features](#).

See [Table 49.34](#) for information on protection of the flash memory area (P/E).

**Table 49.34 Protection for the flash memory area (P/E) (1 of 2)**

FACI command	Target area			Issuing FACI command by Non-secure access	Issuing FACI command by Secure access
Program Block Erase	Code flash memory	User area (Non-secure area)	Secure alias <sup>*1</sup>	X	X
			Non-secure alias <sup>*2</sup>	✓	✓
		User area (Secure area)	Secure alias <sup>*1</sup>	X	✓
			Non-secure alias <sup>*2</sup>	X	X
	Data flash memory	Data area (Non-secure area)	Secure <sup>*1</sup> alias	X	X
			Non-secure alias <sup>*2</sup>	✓	✓
		Data area (Secure area)	Secure alias <sup>*1</sup>	X	✓
			Non-secure alias <sup>*2</sup>	X	X



**Table 49.34 Protection for the flash memory area (P/E) (2 of 2)**

FACI command	Target area			Issuing FACI command by Non-secure access	Issuing FACI command by Secure access
Multi Block Erase Blank Check	Data flash memory	Data area (Non-secure area)	Secure alias <sup>*3</sup>	X	X
			Non-secure alias <sup>*4</sup>	✓	✓
		Data area (Secure area)	Secure alias <sup>*3</sup>	X	✓
			Non-secure alias <sup>*4</sup>	X	X
Configuration set	Code flash memory	Configuration area (Non-secure area)	Secure alias <sup>*1</sup>	X	X
			Non-secure alias <sup>*2</sup>	✓	✓
		Configuration area (Secure area)	Secure alias <sup>*1</sup>	X	✓
			Non-secure alias <sup>*2</sup>	X	X
	Data flash memory	Configuration area (Secure area only)	Secure alias <sup>*1</sup>	X	✓
			Non-secure alias <sup>*2</sup>	X	X
Increment Counter Refresh Counter Read Counter	Data flash memory	Anti-rollback counter area(Secure area only)	-	X	✓

Note:   
 • ✓ indicates FACI command operation is not prohibited.   
 • X indicates FACI command operation is prohibited. Error occurs when the area is selected, and the FACI command is executed.

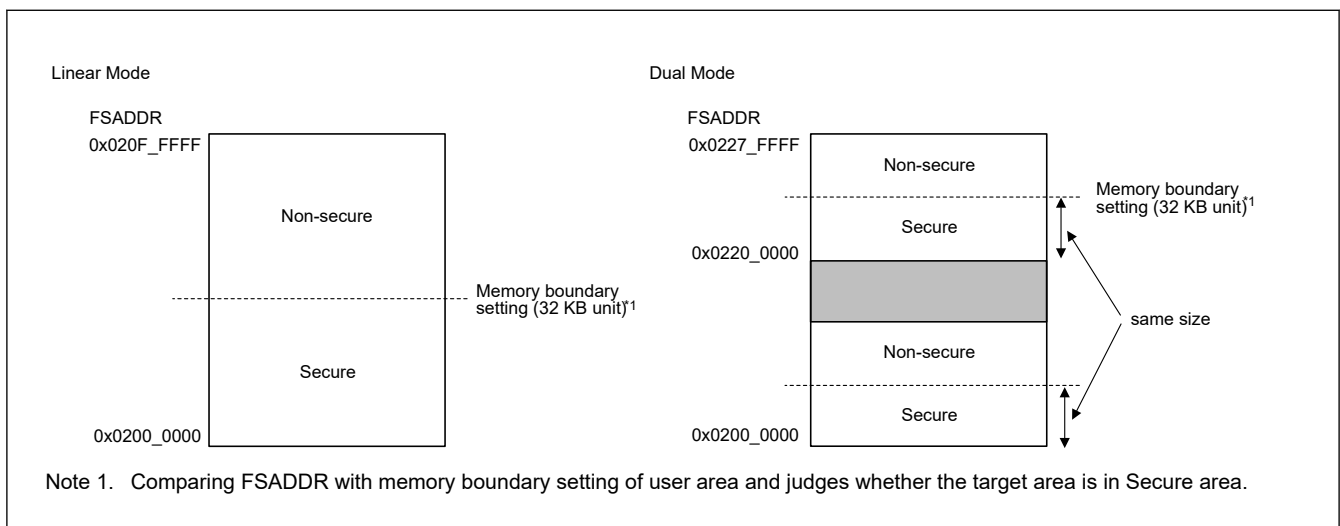
- Note 1. The Secure alias is judged when the address of FSADDR[28] is 0.
- Note 2. The Non-secure alias is judged when the address of FSADDR[28] is 1.
- Note 3. The Secure alias is judged when the address of FSADDR[28] and FEADDR[28] are 0.
- Note 4. The Non-secure alias is judged when the address of FSADDR[28] and FEADDR[28] are 1.

When the target area of FACI command is the user area of code flash, the flash sequencer compares the FSADDR register setting with the memory boundary setting of the code flash and determines whether the target area is in the Secure region.

In linear mode, the memory boundary can be set to 0x0200\_0000 to 0x020F\_8000 in 32 KB unit.

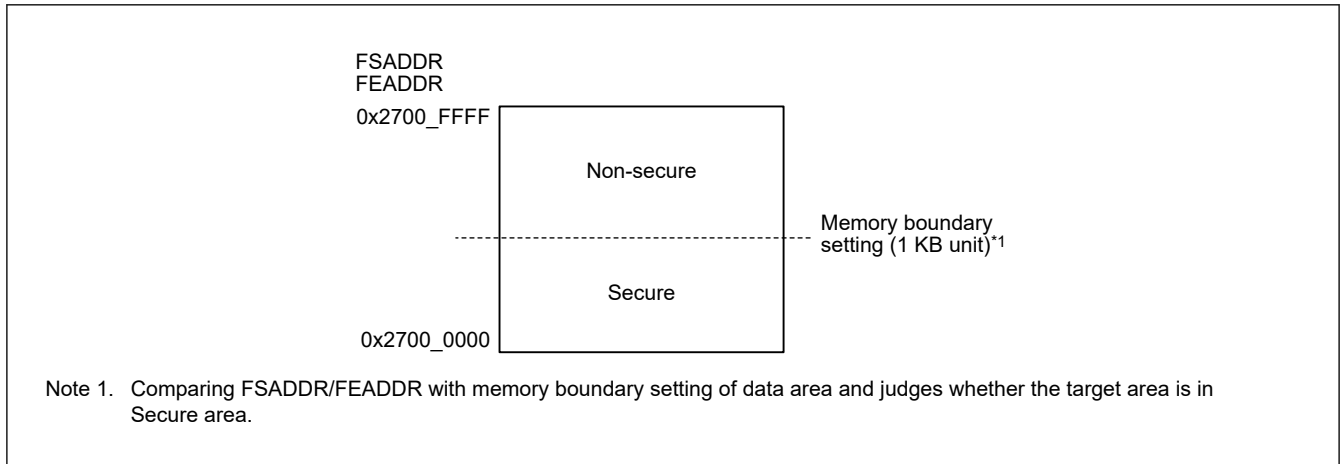
In dual mode, the memory boundary can be set to 0x0200\_0000 to 0x0220\_0000 in 32 KB unit. When the memory boundary is set to 0x0220\_0000 or greater in the dual mode, the entire user area is defined as the Secure region.

Figure 49.39 shows details of the Secure and Non-secure attribute of user area in the code flash.



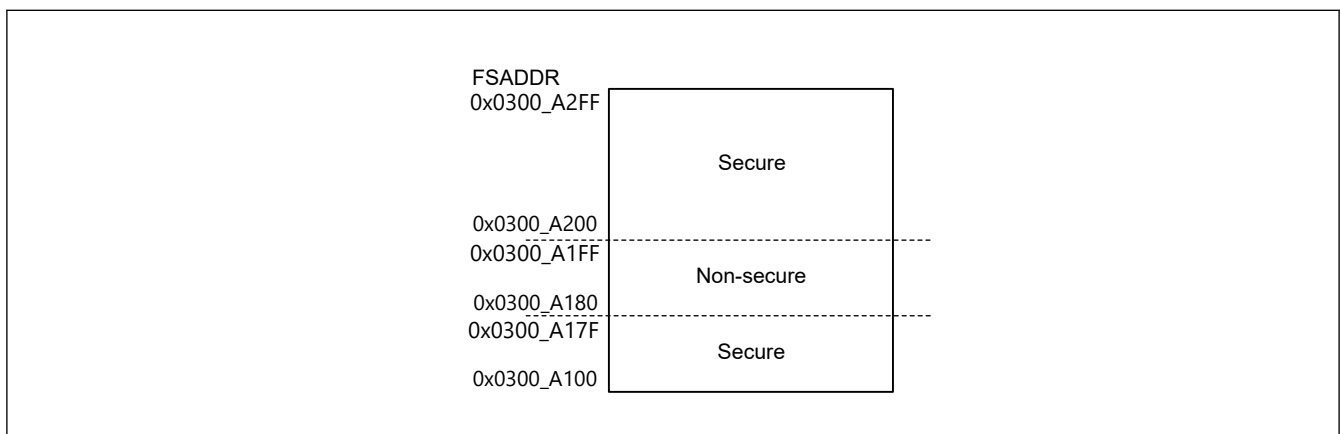
**Figure 49.39 Secure and Non-secure region in user area for Secure alias**

When the target area of the issuing FACI command is the data area of data flash, the flash sequencer compares the FSADDR/FEADDR register setting with the memory boundary setting of the data flash and determines whether the target area is in Secure region. The memory boundary can be set to 0x0270\_0000 to 0x0270\_FC00 in 1 KB unit. Figure 49.40 shows details of the Secure and Non-secure attribute of data area in the data flash.



**Figure 49.40 Secure and Non-secure region in data area for Secure alias**

See Figure 49.41 in the details of Secure and Non-secure region of option-setting memory. The flash sequencer judges that target area is Secure region from the FSADDR register setting.



**Figure 49.41 Secure and Non-secure region in option-setting memory for Secure alias**

### 49.12.3.2 Protection for Flash Memory Area (Read)

This function protects the Secure region of code flash and data flash from Non-secure bus access.

For details of Secure region, see [section 40, Security Features](#) .

### 49.12.3.3 Protection for Register

The flash sequencer registers have write-access protection against Non-secure access. [Table 49.35](#) shows details of the protected registers of the flash sequencer.

**Table 49.35 Protected registers of the flash sequencer for TrustZone (1 of 2)**

Protection target register	Security attribute setting	Notes
FCKMHZ register	SA register setting (FSAR.FCKMHZSA)	For details of <a href="#">section 49.4.4. FSAR : Flash Security Attribution Register</a>
Flash Access area	SA register setting (FSAR.FACICMISA)	For details of <a href="#">section 49.4.4. FSAR : Flash Security Attribution Register</a>

**Table 49.35 Protected registers of the flash sequencer for TrustZone (2 of 2)**

Protection target register	Security attribute setting	Notes
FACI registers* <sup>1</sup> and FWEPROR registers	SA register setting (FSAR.FACICMRSA)	For details of <a href="#">section 49.4.4. FSAR : Flash Security Attribution Register</a>
FCTRCNTR, FCTRLSR FCTRADDR, and FCTRSTATR registers	SA register setting (FSAR.FACITRSA)	For details of <a href="#">section 49.4.4. FSAR : Flash Security Attribution Register</a>
FMEPROT, FCNTSELR, FCNTDATAR0 to 1, FBPROT1 and FSUACR registers	Always Secure	—

Note 1. The target FACI registers are FASTAT, FAEINT, FRDYIE, FSADDR, FEADDR, FBPROT0, FSTATR, FENTRYR, FSUINITR, FCMDR, FBCCNT, FBCSTAT, FPSADDR, FSUASMON, FCPSR, and FPCKAR.

#### 49.12.3.4 Code Flash P/E Mode Entry Protection

The flash sequencer has protection function of code flash P/E by the FMEPROT register for the Secure developer. Secure function can prevent disturbance of reading code flash memory by this protection function. See [section 49.4.14. FMEPROT : Flash P/E Mode Entry Protection Register](#).

For applications that do not require Non-secure region programming/erasure other than from Secure function, it is recommended to always disable Non-secure function of code flash programming/erasure by enabling the protection function of FMEPROT register.

For details, see [Figure 49.42](#) of the code flash P/E sequence example by Non-secure function.

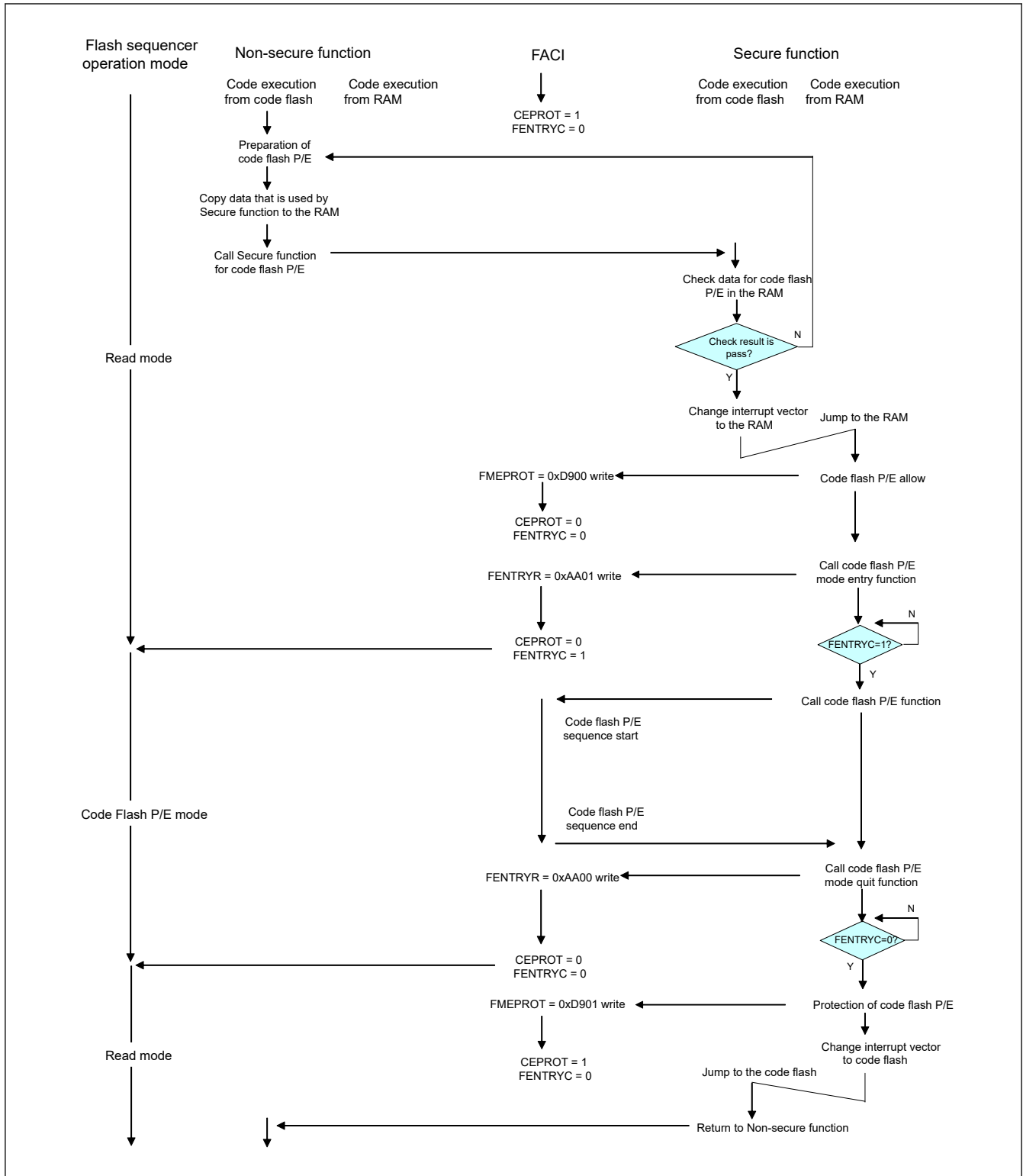


Figure 49.42 Code Flash P/E Sequence Example by Non-secure function (Using Secure function for code flash P/E)

#### 49.12.4 Data Flash Configuration Area Protection

Data flash configuration area has 3 types of lockable area (LK\_CD\_A0, LK\_CD\_A1, and LK\_CD\_A2) by lock bit. When target area of "Configuration set" command includes protection area by lock bit, flash sequencer detects an error and enters "Command Lock" state.

LK\_CD\_A0 and LK\_CD\_A1 are protected 4 bytes data by 1 lock bit.

LK\_CD\_A2 are protected 16 bytes data by 1 lock bit. See in details of protection for flash memory area (P/E). See [Table 49.36](#) in detail of protection condition of lockable area in data flash configuration area.

**Table 49.36 Protection Condition of Lockable area in Data Flash Configuration Area for Secure alias**

Lockable Area		Protection Size	Lock Bit	Remarks
Area Name	address			
LK_CD_A0	0x2703_0098 ~ 0x2703_009B	4 bytes	CD0_LK6	In the case of the 16-byte Configuration set command, data cannot be updated if any of the lock bits corresponding to the update address is 0.
	0x2703_009C ~ 0x2703_009F	4 bytes	CD0_LK7	
	:	:	:	
	0x2703_0170 ~ 0x2703_0173	4 bytes	CD0_LK60	
	0x2703_0174 ~ 0x2703_0177	4 bytes	CD0_LK61	
	0x2703_0178 ~ 0x2703_017B	4 bytes	CD0_LK62	
	0x2703_017C ~ 0x2703_017F	4 bytes	CD0_LK63	
LK_CD_A1	0x2703_0180 ~ 0x2703_0183	4 bytes	CD1_LK0	
	0x2703_0184 ~ 0x2703_0187	4 bytes	CD1_LK1	
	:	:	:	
	0x2703_0270 ~ 0x2703_0273	4 bytes	CD1_LK60	
	0x2703_0274 ~ 0x2703_0277	4 bytes	CD1_LK61	
	0x2703_0278 ~ 0x2703_027B	4 bytes	CD1_LK62	
	0x2703_027C ~ 0x2703_027F	4 bytes	CD1_LK63	
LK_CD_A2	0x2703_0280 ~ 0x2703_028F	16 bytes	CD2_LK0	
	0x2703_0290 ~ 0x2703_029F	16 bytes	CD2_LK1	
	:	:	:	
	0x2703_0340 ~ 0x2703_034F	16 bytes	CD2_LK12	
	0x2703_0350 ~ 0x2703_035F	16 bytes	CD2_LK13	

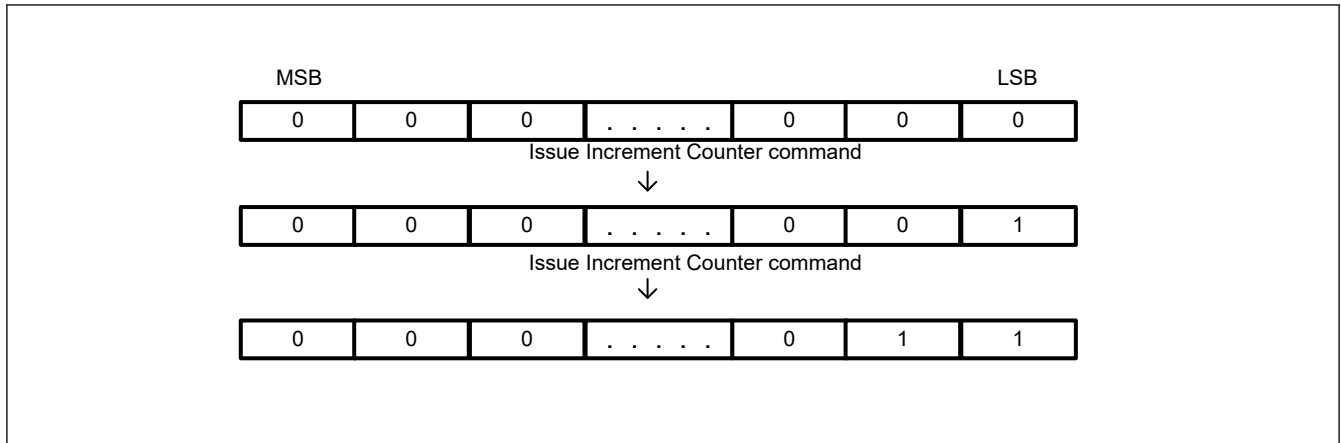
### 49.12.5 Anti-Rollback Counter

See [section 6.2.14. ARCLS : Anti-Rollback Counter Lock Setting](#) to [section 6.2.17. ARC\\_NSECn : Anti-Rollback Counter for Non-secure Application n \(n = 0 to 7\)](#) for Anti-rollback counter related Option flash memory.

Anti-rollback counter cannot be decremented and is used in the firmware update. Only Secure applications are available. Anti-rollback counter shifts one bit to the left with the increment counter command from the initial value of 0. The least significant bit (bit 0) is set to 1. When anti-rollback counter is maximum value, flash sequencer does not change counter by Increment Counter command (does not occur error).

There are three commands that control the counters. For details, see [section 49.9.3.16. Increment Counter Command](#), [section 49.9.3.17. Refresh Counter Command](#), and [section 49.9.3.18. Read Counter Command](#).

[Figure 49.43](#) shows anti-rollback counter increment image when increment counter command is issued.



**Figure 49.43 Increment image of anti-rollback counter**

Anti-rollback counter has 2 types of counters (ARC\_SEC and ARC\_NSEC) and lock bit (ARCSEC\_LK, ARCNSEC\_LK0 to 3) for protection each counter.

ARC\_NSEC can be configured as 1 × 256 or 4 × 64 bits by configuration area setting (CNF\_ARCNS0 and CNF\_ARCNS0) and cannot be updated in invalid setting.

### 49.12.5.1 Anti-Rollback Counter Recovery Flow

Figure 49.44, Figure 49.45 and Figure 49.46-4 shows the recovery flow in the case of anti-rollback counter for secure application. The recovery flow for rest anti-rollback counter should be considered necessary processing to depend on customer's application by Secure software developer. As an example, the flow in dual mode is shown. In linear mode, the same operation can be performed using start-up area select, but application size that be managed by anti-rollback counter should be limited to 8 KB.

### Example Operation: Update from Bank0 selected in Dual mode

**Notation**

UCF: Update Complete Flag set/clear by Secure software.  
 ICF: Increment Complete Flag set/clear by Secure software.  
 Set status: Indicate that update processing is completed.  
 Clear status: Indicate that the update processing is not completed.

User area bank0 (Valid)	Secure application/v0 etc.
User area bank1 (Invalid)	
Data Flash Config. Area	UCF/set
	ICF/set
Code Flash Config. Area	Bank0 select
Anti-rollback counter	Counter/v0

**Step 0 Status before update**

Secure application is before version update in bank0.



User area bank0 (Valid)	Secure application/v0 etc.
User area bank1 (Invalid)	<b>Secure application/v1 etc.</b>
Data Flash Config. Area	<b>UCF/clear</b>
	<b>ICF/clear</b>
Code Flash Config. Area	Bank0 select
Anti-rollback counter	Counter/v0

**Step 1 Write new secure application, Clear UCF and ICF**

UCF and ICF are cleared by configuration set command.  
 Next, new secure application etc. is written to invalid user area by program/erase command.

**How to select recovery flow:**

It can check that power failure is occurred by below status.

- UCF is clear status.
- Counter value is old version.

**Recovery flow:**

It should resume from step1.



**Figure 49.44 Recovery flow of secure application update in Dual mode at power failure (1/3)**

### Example Operation: Update from Bank0 selected in Dual mode

**Notation**

UCF: Update Complete Flag set/clear by Secure software.  
 ICF: Increment Complete Flag set/clear by Secure software.  
 Set status: Indicate that update processing is completed.  
 Clear status: Indicate that the update processing is not completed.

User area bank0 (Valid)	Secure application/v0 etc.
User area bank1 (Invalid)	Secure application/v1 etc.
Data Flash Config. Area	<b>UCF/set</b>
	ICF/clear
Code Flash Config. Area	Bank0 select
Anti-rollback counter	Counter/v0

**Step 2 Set UCF**

UCF is set by configuration set command.

**How to select recovery flow:**

Same as step1.

**Recovery flow:**

Same as step1.



User area bank0 (Valid)	Secure application/v0 etc.
User area bank1 (Invalid)	Secure application/v1 etc.
Data Flash Config. Area	UCF/set
	<b>ICF/set</b>
Code Flash Config. Area	Bank0 select
Anti-rollback counter	<b>Counter/v1</b>

**Step 3 Update counter version, Set ICF**

Secure application version counter is update by increment counter command. Next, ICF is set by configuration set command.

**How to select recovery flow(a):**

It can check that power failure is occurred by below status.

- UCF is set status.
- ICF is clear status.
- Counter value is old version.

**Recovery flow(a):**

It should issue refresh counter command. Next, after clearing UCF, it should resume from step1.

**How to select recovery flow(b):**

It can check that power failure is occurred by below status.

- UCF is set status.
- ICF is clear status.
- Counter value is new version.

**Recovery flow(b):**

It should issue refresh counter command and set ICF. Next, it should resume from step4.



User area bank0 (Valid)	Secure application/v0 etc.
User area bank1 (Invalid)	Secure application/v1 etc.
Data Flash Config. Area	UCF/clear
	ICF/clear
Code Flash Config. Area	<b>Bank1 select</b>
Anti-rollback counter	Counter/v1

**Step 4 Update bank selection**

BANKSWP[2:0] bits are changed to bank1 selection by configuration set command.

**How to select recovery flow:**

It can check that power failure is occurred by below status.

- UCF is set status.
- ICF is set status.
- Counter value is new version.
- BANKSWP[2:0] bits are bank0 selection.

**Recovery flow:**

It should resume from step4.



Figure 49.45 Recovery flow of secure application update in Dual mode at power failure (2/3)



### Example Operation: Update from Bank0 selected in Dual mode

#### Notation

UCF: Update Complete Flag set/clear by Secure software.  
 ICF: Increment Complete Flag set/clear by Secure software.  
 Set status: Indicate that update processing is completed.  
 Clear status: Indicate that the update processing is not completed.

User area bank0 (Invalid)	Secure application/v0 etc.
User area bank1 (valid)	Secure application/v1 etc.
Data Flash Config. Area	UCF/set
	ICF/clear
Code Flash Config. Area	Bank1 select
Anti-rollback counter	Counter/v1

#### Step 5 Switch valid user area

Bank1 is changed to valid area after system reset.  
 New secure application should check that update processing is completed successfully by checking UCF, ICF, and anti-rollback counter

Figure 49.46 Recovery flow of secure application update in Dual mode at power failure (3/3)

## 49.13 Boot Mode

There are two serial programming modes; the boot mode (for the SCI interface) with SCI9 and the boot mode (for the USB interface) with USBFS. Table 49.37 lists the I/O pins used in boot mode. Table 49.38 lists the available communication interface used in the boot mode.

Table 49.37 I/O Pins Used in Boot Mode

Pin Name	I/O	Mode to be Used	Use
MD	Input	Boot mode (for the SCI interface) Boot mode (for the USB interface) Boot mode (for the JTAG interface) Boot mode (for the SWD interface)	Selection of operating mode
P208/RXD9	Input	Boot mode(for the SCI interface)	For host communication (to receive data through SCI)
P209/TXD9	Output		For host communication (to transmit data through SCI)
USB_DP, USB_DM	I/O	Boot mode(for the USB interface)	Data input/output of USB
USB_VBUS	Input		Detection of connection and disconnection of USB cables
P211/TCK	Input	Boot mode (for the JTAG interface)	Clock pin of JTAG
P210/TMS	Input		Mode Select pin of JTAG
P208/TDI	Input		Data In pin of JTAG
P209/TDO	Output		Data Out pin of JTAG
P211/SWCLK	Input	Boot mode (for the SWD interface)	Clock pin of SWD
P210/SWDIO	I/O		Data In-Out pin of SWD

Table 49.38 Available Communication Interface Used in Boot Mode (1 of 2)

	Yes	No	No
Main clock oscillator or external clock is connected	Yes	No	No
Sub clockoscillator is connected*1	Yes or No	Yes	No
Available interface	JTAG, SWD, SCI or USB	JTAG, SWD, SCI or USB	JTAG, SWD or SCI

**Table 49.38 Available Communication Interface Used in Boot Mode (2 of 2)**

Tool connection time*2	Up to 1 second	Up to 2 seconds	Up to 3 seconds
------------------------	----------------	-----------------	-----------------

Note 1. The drive capability of the sub clock oscillator is set to standard by SOMCR.SODRV bit. Note that if you use the crystal corresponding the low drive capability on your board, the crystal may not oscillate in the boot mode.

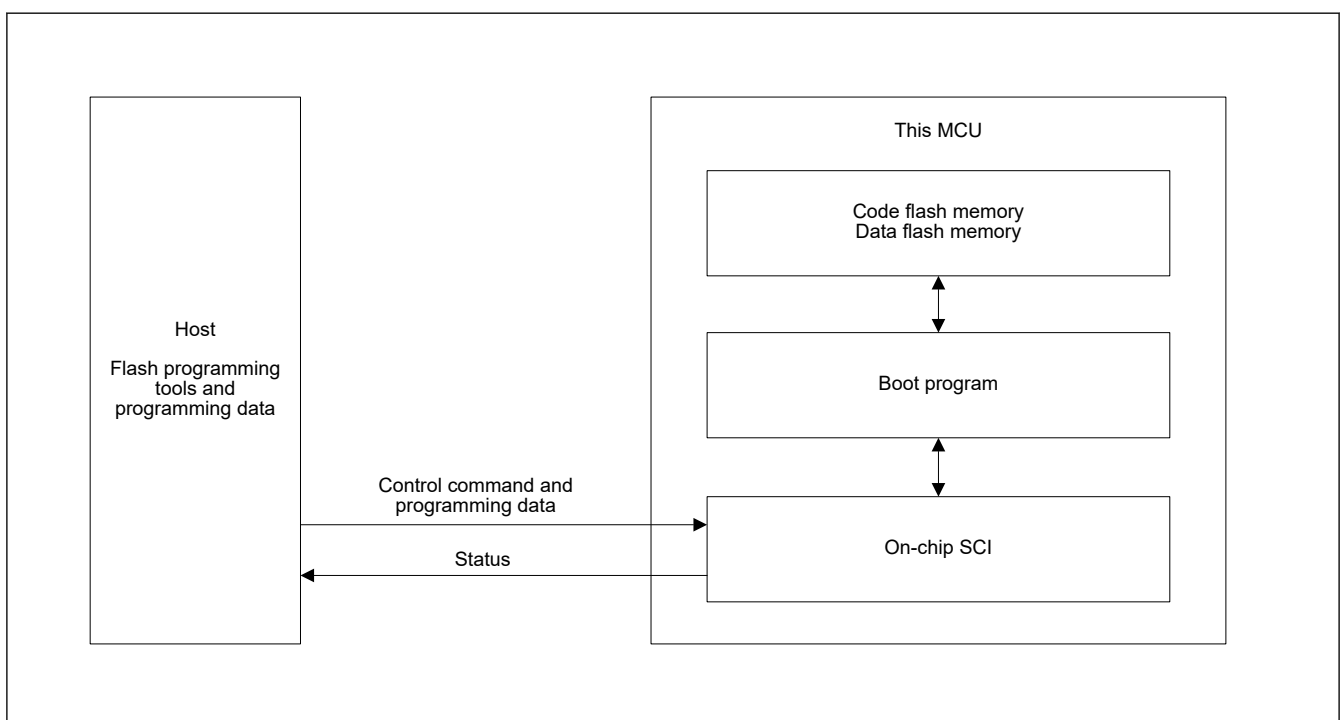
Note 2. See the boot firmware manual for the detail connection time.

### 49.13.1 Boot Mode (for the SCI Interface)

In boot mode (for the SCI interface), the host sends control commands and data for programming, and the flash memory is programmed or erased accordingly. An on-chip SCI handles transfer between the host and this MCU in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.

When this MCU is activated in boot mode (for the SCI interface), the program on the dedicated area the MCU is executed. The boot program automatically adjusts the bit rate of the SCI and controls programming/erasure by receiving control commands from the host.

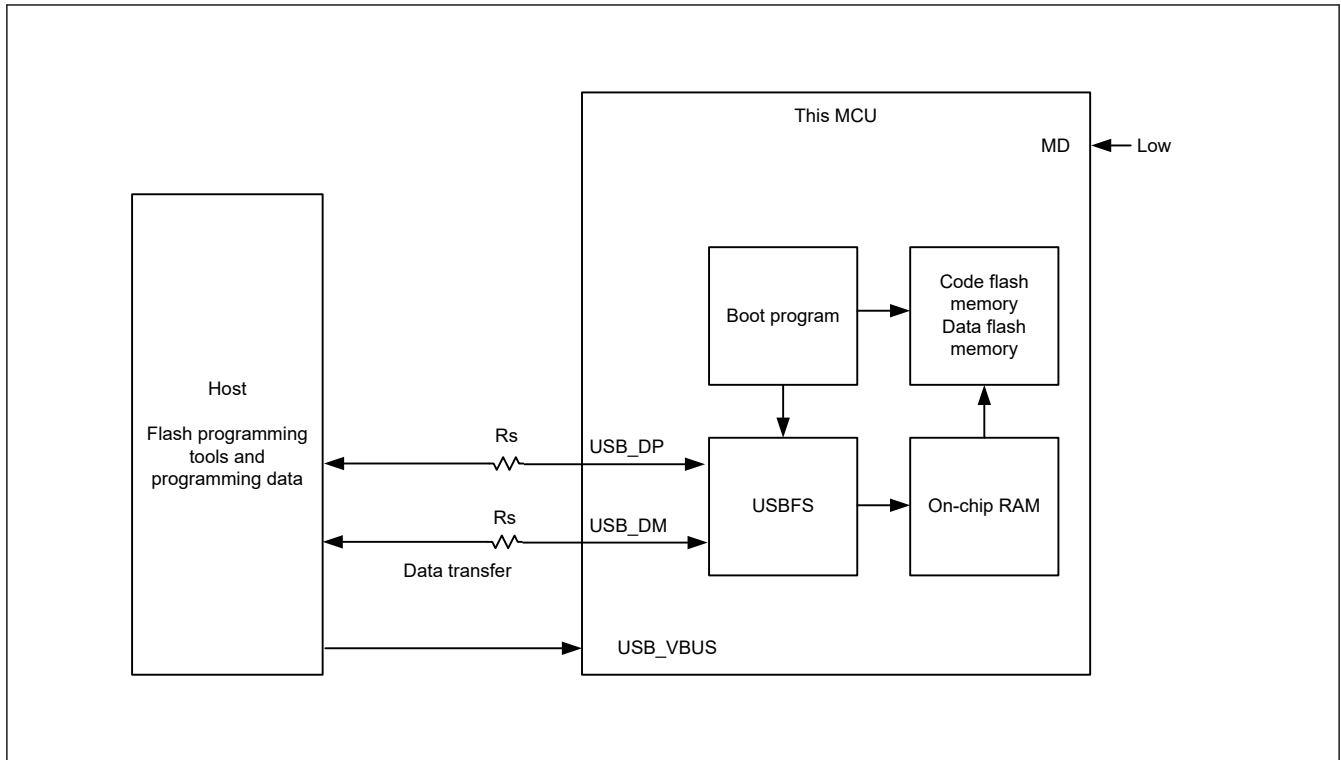
Figure 49.47 shows the system configuration for operations in boot mode (for the SCI interface).



**Figure 49.47 System Configuration for Operations in Boot Mode (for the SCI Interface)**

### 49.13.2 Boot Mode (for the USB Interface)

In boot mode (with the USB interface), the flash memory can be programmed or erased by sending control commands and program data from the host. An on-chip USB is used for communications between the host and this MCU. The host requires tools for sending control commands and data for programming. Figure 49.48 shows the configuration of a system for use in boot mode (for the USB interface). The USB cable must be connected on reset release.



**Figure 49.48 System Configuration in Boot Mode (for the USB Interface)**

## 49.14 Using the Serial Programmer for Rewriting

A serial programmer can be used to rewrite flash memory in boot mode.

### (1) Serial Programming

This MCU is mounted on the system board at the time of serial programming. Providing a connector to the board enables rewriting of this MCU by the serial programmer to proceed.

#### 49.14.1 Environments for Serial Programming

The recommended environments for rewriting the flash memory of the MCU with data are described below.

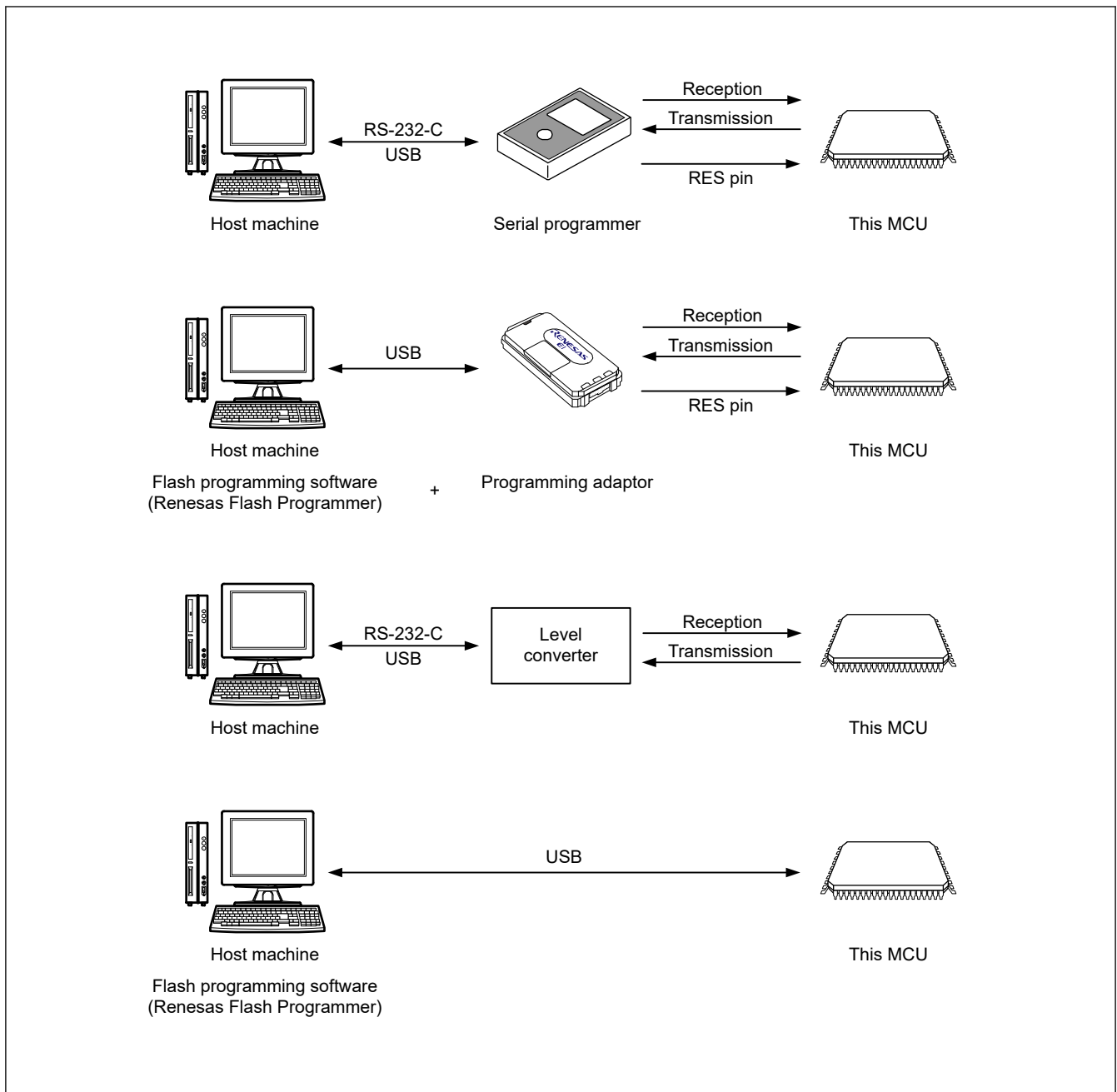


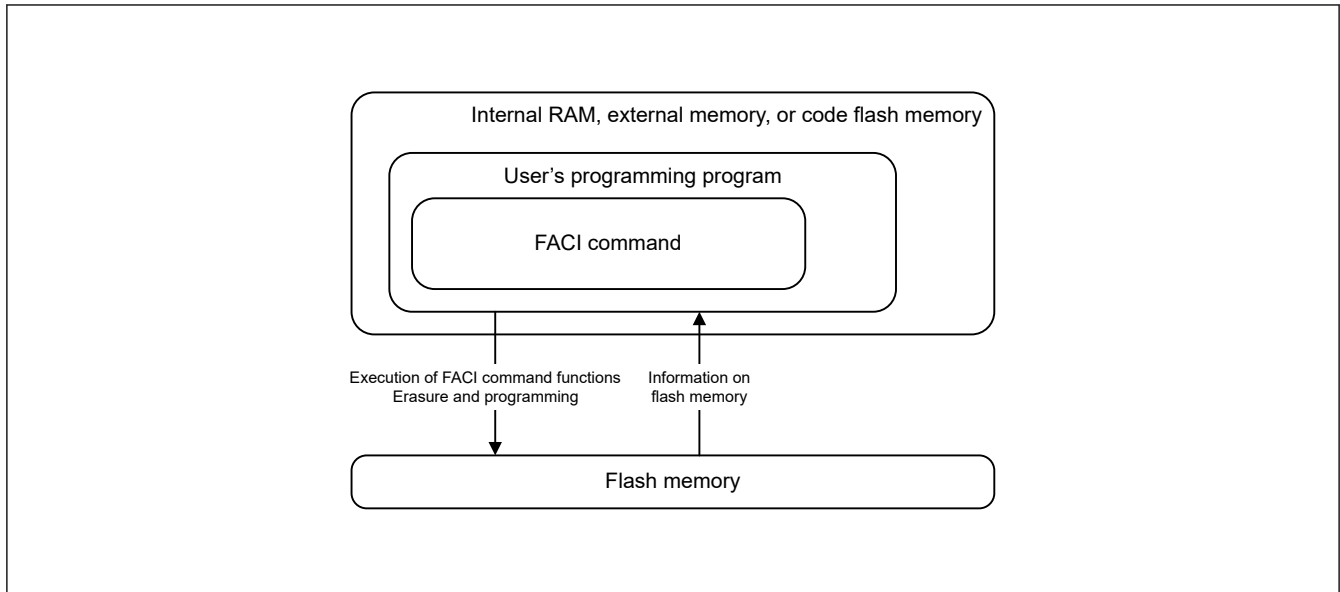
Figure 49.49 Environments for Rewriting the Flash Memory

## 49.15 Programming through Self-Programming

### 49.15.1 Overview

This MCU supports programming of the flash memory by the user program itself. The FACL commands can be used with user programs for writing to the flash memory. This allows upgrading of user programs and rewriting of constant data fields.

The program for rewriting must be transferred to the internal RAM or external memory in advance when the BGO is not available or when rewriting the option-setting memory.



**Figure 49.50 Schematic View of Self-Programming**

For comprehensive information on the self-programming, see [section 49.9. FACL Commands](#).

### 49.15.2 Background Operation

The background operation (BGO) can be used to execute the flash rewrite routine on the code flash memory when the data flash memory or other area of code flash memory is rewritten.

Background operations can be used when the combination of the flash memory for rewriting and the flash memory for reading is any of those listed below.

**Table 49.39 Conditions under which Background Operation is Usable**

		Range for rewriting	Range for reading
Common to linear and dual modes		Code flash memory	Data flash memory
		Data flash memory	Code flash memory
Liner mode	Products with 1 Mbytes of code flash memory	First half (0.5 Mbytes) of the code flash memory addresses Secure alias: 0x0200_0000 to 0x0207_FFFF Non-secure alias: 0x1200_0000 to 0x1207_FFFF	Second half (0.5 Mbytes) of the code flash memory addresses Secure alias: 0x0208_0000 to 0x020F_FFFF Non-secure alias: 0x1208_0000 to 0x120F_FFFF
		Second half (0.5 Mbytes) of the code flash memory addresses Secure alias: 0x0208_0000 to 0x020F_FFFF Non-secure alias: 0x1208_0000 to 0x120F_FFFF	First half (0.5 Mbytes) of the code flash memory addresses Secure alias: 0x0200_0000 to 0x0207_FFFF Non-secure alias: 0x1200_0000 to 0x1207_FFFF
Dual mode	When the BANKSEL_SEC.BANKSWP[2:0] or BANKSEL.BANKSWP[2:0] bits are 111b	Bank 1 area of the code flash memory	Bank 0 area of the code flash memory
	When the BANKSEL_SEC.BANKSWP[2:0] or BANKSEL.BANKSWP[2:0] bits are 000b	Bank 0 area of the code flash memory	Bank 1 area of the code flash memory

## 49.16 Reading Flash Memory

### 49.16.1 Reading Code Flash Memory

Special settings are not required to read code flash memory after release from the reset state. Data can simply be read out through access to addresses in the code flash memory.

When reading code flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state), all bits are read as 1.

### 49.16.2 Reading Data Flash Memory

Special settings are not required to read data flash memory after release from the reset state. Data can simply be read out through access to addresses in the data flash memory.

Values read from data flash memory that has been erased but not yet been programming again (i.e. that is in the non-programmed state) are undefined. Use blank checking when you need to confirm that an area is in the non-programmed state.

### 49.17 Usage Notes

#### (1) Reading Area Where Programming/Erase was Interrupted and Area Targeted for Suspension

The data stored in the area where programming or erasure has been suspended or the area where programming or erasure has been suspended by using the suspend command are undefined. To avoid faulty operation caused by reading undefined data, take care not to fetch instructions or read data from areas where programming or erasure was suspended and where programming or erasure was suspended by using the suspend command.

#### (2) Suspension During Programming/Erase

When processing of programming/erasure is stopped by issuing the P/E suspend command, the programming/erasure processing can be resumed by issuing the P/E resume command. If the flash sequencer enters the command-locked state for any reason and issues the forced stop command after the suspended processing is normally completed and the ERSSPD flag or PRGSPD flag is set to 1, the suspended processing cannot be resumed. In addition, the values in the area where the processing was suspended are not guaranteed. Erase that area

#### (3) Prohibition of Additional Programming

Programming a given area of the code flash memory or data flash memory twice is not possible. To program the code flash memory or data flash memory where has been programmed, erase the target area. Programming can be added to the option-setting memory.

#### (4) Resets During Programming/Erase, or Blank Checking

In the case of a reset due to the signal on the RES pin during programming/erasure, or blank checking of the flash memory, wait for at least  $t_{RESW}$  (see [section 52, Electrical Characteristics](#)) of the reset input period once the operating voltage is within the range stipulated in the electrical characteristics, then release the device from the reset state.

#### (5) Allocation of Vectors for Interrupts and Other Exceptions During Programming/Erase

Generation of an interrupt or other exception during programming/erasure may lead to fetching of the vector from the code flash memory. Under conditions where BGO cannot be used, set the address of the vector to an address that is not in the code flash memory. Alternatively, make sure that no handling of interrupts or exceptions proceeds during programming/erasure.

#### (6) Items Prohibited During Programming/Erase, or Blank Checking

High voltage is applied to the flash memory during programming/erasure, or blank checking. To prevent damage to the flash memory, do not perform the following operations.

- Have the operating voltage from the power supply go beyond the permitted range.
- Change the FWEPROR.FLWE[1:0]bits.
- Change the OPCCR.OPCM[2:0] bits.
- Change the SCKDIVCR.FCK[2:0]bits.
- Change the SCKSCR.CKSEL[2:0]bits.
- Transition to the software standby mode, or Deep Software Standby mode.

#### (7) Programming/Erase in Low-Speed Modes

Do not programming/erasure the flash memory when low-speed mode is selected with the operating power control register (OPCCR).

(8) Setting dual bank mode and programming in Boot Mode

The initial mode of the MCU shipped from Renesas is linear mode. In customer's factory when programming dual bank mode and customer's applications on MCU set in linear mode, it is recommended that only area1 in Figure 49.51 is programmed in boot mode and area2 is kept blank. After reset, MCU starts in dual mode and the application is in bank0. Use self-programming when programming area2 for updating in the field. See section 40.7. Field Updating in Dual Mode for details

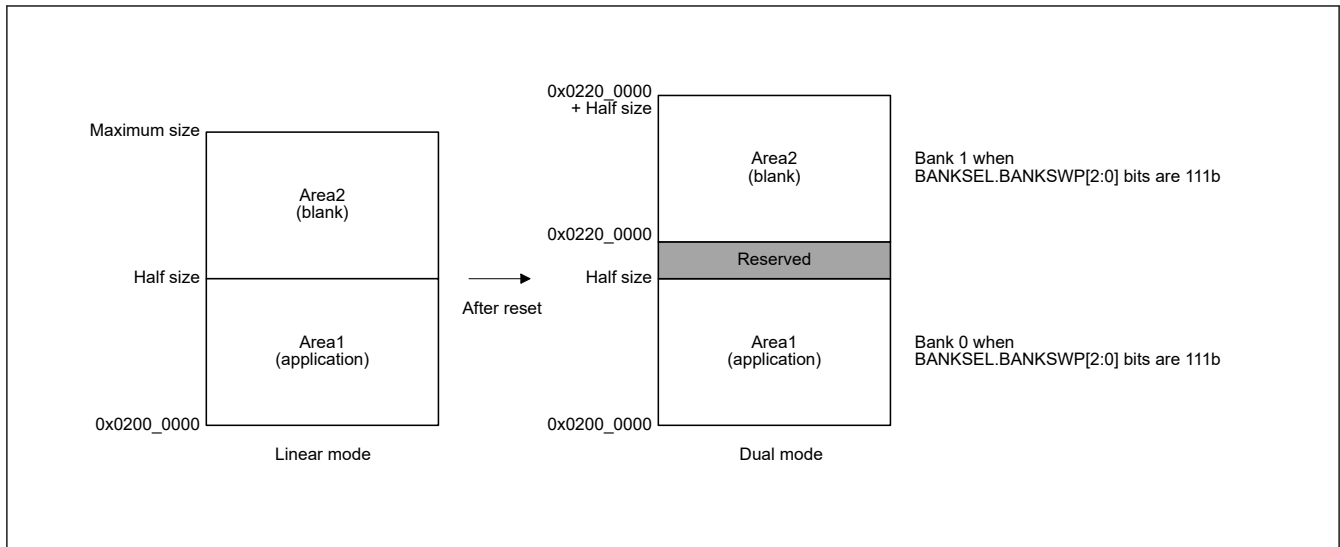


Figure 49.51 Programming the application for the dual mode in the linear mode

(9) Emulator Connection

Renesas provides the emulator which supports both debugging using SWD or JTAG communication and serial programming using SCI communication.

Table 49.40 shows the pinout of 10 pin or 20 pin socket pinouts when using this emulator. The pinout of SWD and JTAG is ARM standard, and MD, TXD, RXD pins are added for the serial programming using SCI communication.

Table 49.40 Pin assign for emulator

Pin No.	SWD	JTAG	Serial Programming using SCI
1	VCC	VCC	VCC
2	P210/SWDIO	P210/TMS	NC
4	P211/SWCLK	P211/TCK	P201/MD
6	P209/SWO	P209/TDO	P209/TXD9
8	P208	P208/TDI	P208/RXD9
9	GNDdetect	GNDdetect	GNDdetect
10	nRESET	nRESET	nRESET
12	P308/TCLK	P308/TCLK	NC
14	P307/TDATA[0]	P307/TDATA[0]	NC
16	P306/TDATA[1]	P306/TDATA[1]	NC
18	P305/TDATA[2]	P305/TDATA[2]	NC
20	P304/TDATA[3]	P304/TDATA[3]	NC
3, 5, 15, 17, 19	GND	GND	GND
7	NC	NC	NC
11, 13	NC	NC	NC

## 50. Capture Engine Unit (CEU)

### 50.1 Overview

The Capture Engine Unit (CEU) is a capture module that fetches image data externally input and transfers it to the memory. The CEU is connected to the system bus via bus bridge modules.

Lists the features of CEU as follows.

#### (1) Image data fetch

- Captures an image output from an external module and writes YCbCr data to the memory with it separated into Y data and CbCr data.
- Fetches image data other than YCbCr data, for example, JPEG data, RGB565, from an externally connected module, such as a camera, and sequentially writes the image data to the memory.
- Fetches an interlace source image in both-field units or one-field units and writes it to the memory. In both-field capture, an image can be stored in the memory as a frame image.

#### (2) Filter processing

- Performs scale-down and removal of high-frequency components (only in the horizontal direction) for an image using internal filters. Note that the scaled-down image must not exceed VGA. The filter processing can be applied to only YCbCr input data.

#### (3) Format conversion

- Converts image data input in the YCbCr422 format into the YCbCr420 format and writes it to the memory. Note that the conversion algorithm is simple thinning in which the chrominance component (CbCr) of the evennumbered lines is thinned.

The functional overview of the CEU is shown in [Table 50.1](#) and the main functions and their details are shown in [Table 50.2](#).



Table 50.1 Functional overview of CEU (1 of 2)

Classification	Item	Function	Description	Note	
Connectable camera	Size <sup>*1</sup>	5 megapixels	2,560 pixels × 1,920 lines	Image capture mode or data synchronous fetch mode Horizontal : 8-pixel units Vertical : 4-line units Data enable fetch mode 32-byte units The range of the image size that can be input is as follows. Horizontal: 2,560 pixels to 128 pixels Vertical: 1,920 pixels lines to 96 lines	
		3 megapixels	2,048 pixels × 1,536 lines		
		2 megapixels	1,632 pixels × 1,224 lines		
		UXGA	1,600 pixels × 1,200 lines		
		SXGA (1)	1,280 pixels × 1,024 lines		
		SXGA (2)	1,280 pixels × 960 lines		
		WXGA	1,280 pixels × 768 lines		
		XGA	1,024 pixels × 768 lines		
		SVGA	800 pixels × 600 lines		
		WVGA	800 pixels × 480 lines		
		VGA	640 pixels × 480 lines		
		CIF	352 pixels × 288 lines		
		WQVGA	480 pixels × 240 lines		
		QVGA	320 pixels × 240 lines, 240 pixels × 320 lines		
		QCIF	176 pixels × 144 lines		
	QQVGA	160 pixels × 120 lines			
	Sub-QCIF	128 pixels × 96 lines			
	Input format	YCbCr422 8 bits		Cb <sub>0</sub> , Y <sub>0</sub> , Cr <sub>0</sub> , Y <sub>1</sub> ...	Supports clock ratio of 1:1
				Cr <sub>0</sub> , Y <sub>0</sub> , Cb <sub>0</sub> , Y <sub>1</sub> ...	
				Y <sub>0</sub> , Cb <sub>0</sub> , Y <sub>1</sub> , Cr <sub>0</sub> ...	
Y <sub>0</sub> , Cr <sub>0</sub> , Y <sub>1</sub> , Cb <sub>0</sub> ...					
Binary data			Specified amount to be fetched on edges of the sync signal Data is fetched with the horizontal sync signal as an enable signal	Written sequentially	
Horizontal and vertical sync signal polarities	Arbitrary	High-active and low-active			
Capture start location	Arbitrary	Can be specified in camera input clock units	Horizontal: 1-cycle units Vertical: 1-HD (horizontal sync signal) units		
Number of captured pixels	Arbitrary	Can be specified in 8-pixel units horizontally and in 4-line units vertically			
Interlace	Both-field capture	Stored as a field image	Capture: 2-VD (vertical sync signal) units		
		Stored as a frame image			
	One-field capture	Top field or bottom field can be specified	Capture: 1-VD units		
Memory write	Output format	YCbCr422 YCbCr420	YCbCr420 is realized by simple thinning		

**Table 50.1 Functional overview of CEU (2 of 2)**

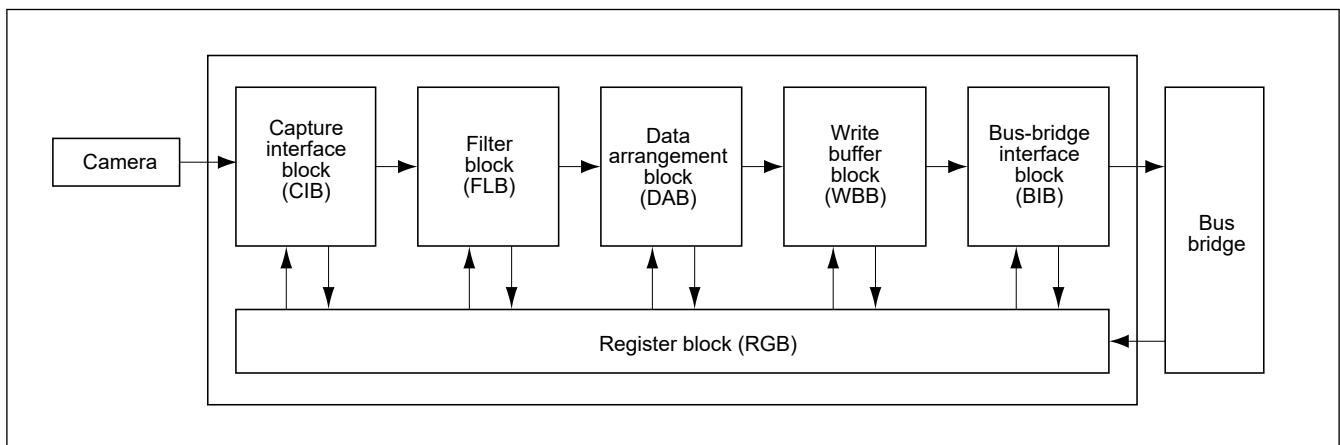
Classification	Item	Function	Description	Note
Filter function	No scaling or scale-down	Scale-down of captured display	Desired scaling factor from 1/16 to 1 (scaled-down display must not exceed VGA)	
	Low-pass filter		Removal of high-frequency components	Only in the horizontal direction
Module-stop function	Module-stop state can be set to reduce power consumption.			
TrustZone Filter	Security and Privilege attribution can be set.			

Note 1. The range of the image size that can be input depends on the AC characteristics and the frame rate of the connected device, and the transfer speed to the destination RAM.

**Table 50.2 Main functions of CEU and their details**

Main function	Detailed description
Image data fetch	<ul style="list-style-type: none"> <li>• Captures an image output from an external module and writes YCbCr data to the memory with it separated into Y data and CbCr data.</li> <li>• Fetches image data other than YCbCr data, e.g. JPEG data, from an externally connected module, such as a camera, and sequentially writes the image data to the memory.</li> <li>• Fetches an interlace source image in both-field units or one-field units and writes it to the memory. In both-field capture, an image can be stored in the memory as a frame image.</li> </ul>
Filter processing	Performs scale-down and removal of high-frequency components (only in the horizontal direction) for an image using internal filters. Note that the scaled-down image must not exceed VGA. The filter processing can be applied to only YCbCr input data.
Format conversion	Converts image data input in the YCbCr422 format into the YCbCr420 format and writes it to the memory. Note that the conversion algorithm is simple thinning in which the chrominance component (CbCr) of the even-numbered lines is thinned.

Figure 50.1 shows a block diagram of the CEU.



**Figure 50.1 Block diagram of CEU**

The pin configuration of the CEU is shown in Table 50.3.

**Table 50.3 Pin configuration of CEU (1 of 2)**

Pin name	Function	I/O	Description
VIO_D7 to VIO_D0*1	CEU data bus	Input	Camera image data input to the CEU
VIO_CLK	CEU clock	Input	Camera clock input to the CEU
VIO_VD	CEU vertical sync	Input	Camera vertical sync signal input to the CEU

**Table 50.3 Pin configuration of CEU (2 of 2)**

Pin name	Function	I/O	Description
VIO_HD	CEU horizontal sync	Input	Camera horizontal sync signal input to the CEU
VIO_FLD <sup>*2</sup>	Field signal	Input	Field identification signal to the CEU

Note 1. When the distinction according to the bus width for the data bus is not needed, VIO\_D is used in this manual. Otherwise, VIO\_D7 to VIO\_D0 are used.

Note 2. The VIO\_FLD pin is valid only in interlace input mode (CAIFR.IFS=1).

## 50.2 Register Descriptions

The register configuration of the CEU is shown in [Table 50.4](#).

Most CEU registers have a 2-plane configuration (plane A and plane B). The CEU switches the planes when using these 2-plane registers. A mirror address, which is an address that can always access the register on the unused plane, is provided for each 2-plane register. [Figure 50.2](#) shows the timing to switch the register planes. The CEU switches the register planes at the same time a VD interrupt is asserted.

In the following register descriptions, "during operation" indicates the period that begins when the CEU is activated by the CE bit in the capture start register (CAPSR) and ends when a capture end interrupt (CPE) of the capture event flag clear register (CETCR) occurs. In the read-only bits in each register, the write value should always be 0. If a value other than 0 is written to any of these bits, correct operation cannot be guaranteed.

**Table 50.4 Register configuration of CEU (1 of 2)**

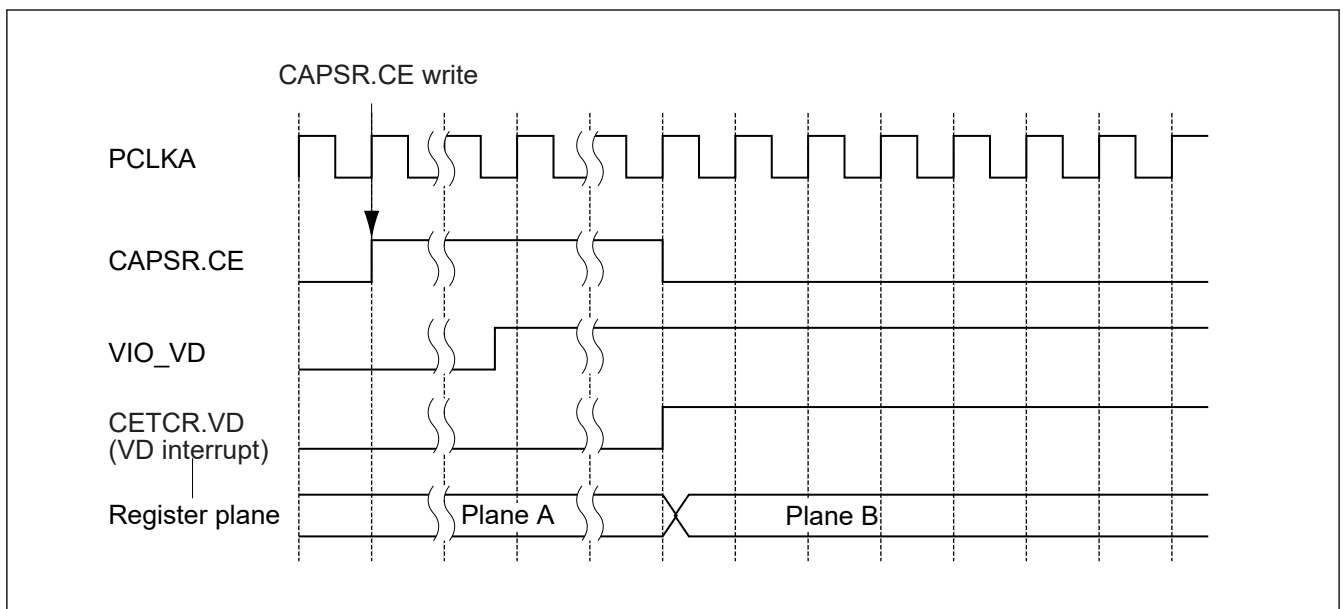
Register name	Abbr.	R/W	Addresses			Access size
			Address (Plane A)	Address (Plane B)	Mirror Address	
Capture Start Register	CAPSR	R/W	0x4034_8000	—	—	32
Capture Control Register	CAPCR	R/W	0x4034_8004	—	—	32
Capture Interface Control Register <sup>*1</sup>	CAMCR	R/W	0x4034_8008	—	—	32
Capture Interface Cycle Register <sup>*1</sup>	CMCYR	R/W	0x4034_800C	—	—	32
Capture Interface Offset Register	CAMOR	R/W	0x4034_8010	0x4034_9010	0x4034_A010	32
Capture Interface Width Register	CAPWR	R/W	0x4034_8014	0x4034_9014	0x4034_A014	32
Capture Interface Input Format Register	CAIFR	R/W	0x4034_8018	—	—	32
CEU Register Control Register	CRCNTR	R/W	0x4034_8028	—	—	32
CEU Register Forcible Control Register	CRCMPR	R/W	0x4034_802C	—	—	32
Capture Filter Control Register	CFLCR	R/W	0x4034_8030	0x4034_9030	0x4034_A030	32
Capture Filter Size Clip Register	CFSZR	R/W	0x4034_8034	0x4034_9034	0x4034_A034	32
Capture Destination Width Register	CDWDR	R/W	0x4034_8038	0x4034_9038	0x4034_A038	32
Capture Data Address Y Register	CDAYR	R/W	0x4034_803C	0x4034_903C	0x4034_A03C	32
Capture Data Address C Register	CDACR	R/W	0x4034_8040	0x4034_9040	0x4034_A040	32
Capture Data Bottom-Field Address Y Register	CDBYR	R/W	0x4034_8044	0x4034_9044	0x4034_A044	32
Capture Data Bottom-Field Address C Register	CDBCR	R/W	0x4034_8048	0x4034_9048	0x4034_A048	32
Capture Bundle Destination Size Register	CBDSR	R/W	0x4034_804C	0x4034_904C	0x4034_A04C	32
Firewall Operation Control Register	CFWCR	R/W	0x4034_805C	—	—	32
Capture Low-Pass Filter Control Register	CLFCR	R/W	0x4034_8060	0x4034_9060	0x4034_A060	32
Capture Data Output Control Register	CDOCR	R/W	0x4034_8064	0x4034_9064	0x4034_A064	32
Capture Event Interrupt Enable Register	CEIER	R/W	0x4034_8070	—	—	32
Capture Event Flag Clear Register	CETCR	R/W	0x4034_8074	—	—	32

**Table 50.4 Register configuration of CEU (2 of 2)**

Register name	Abbr.	R/W	Addresses			Access size
			Address (Plane A)	Address (Plane B)	Mirror Address	
Capture Status Register	CSTSR	R	0x4034_807C	—	—	32
Capture Data Size Register	CDSSR	R/W	0x4034_8084	—	—	32
Capture Data Address Y Register 2	CDAYR2	R/W	0x4034_8090	0x4034_9090	0x4034_A090	32
Capture Data Address C Register 2	CDACR2	R/W	0x4034_8094	0x4034_9094	0x4034_A094	32
Capture Data Bottom-Field Address Y Register 2	CDBYR2	R/W	0x4034_8098	0x4034_9098	0x4034_A098	32
Capture Data Bottom-Field Address C Register 2	CDBCR2	R/W	0x4034_809C	0x4034_909C	0x4034_A09C	32
CEU Bufferable Write Enable Register	CBWER	R/W	0x4034_80A0	—	—	32

Note: The address shown in this table are Secure region. The address corresponding to the Non-secure region is when b28 of address is set to 1.

Note 1. After changing the setting of a register (CAMCR or CMCYR) that is determined by the external module characteristics, do not start capture until at least 10 external input clock cycles have elapsed.



**Figure 50.2 Register plane switching timing (VD polarity is high-active in data enable fetch mode)**

### 50.2.1 CAPSR : Capture Start Register

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0000

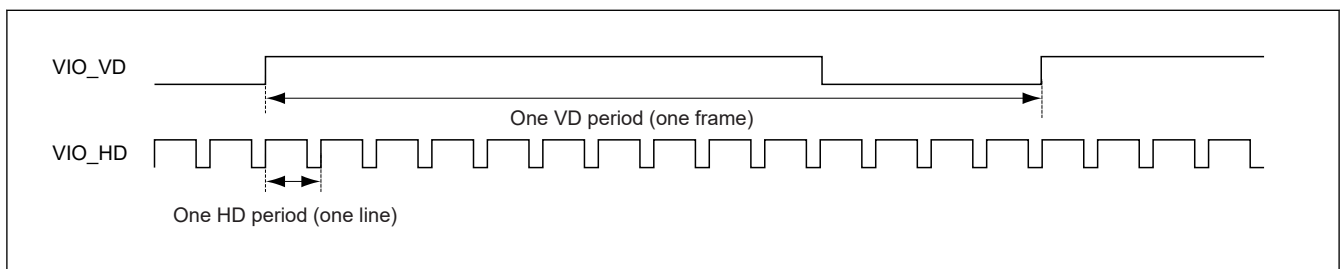
Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPKIL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CE	Capture enable <ul style="list-style-type: none"> <li>• Single-capture This bit reserves capturing of the next frame. When 1 is written to this bit, the capture of one frame starts from the next VD input, and stops when the one-frame capture end interrupt (CPE bit in CETCR) is asserted (Figure 50.7). To perform capture again, write 1 to this bit. After the VD or HD polarity is changed, do not write 1 to this bit until the next VD interrupt is asserted. As this bit indicates the capture reserve state, this bit is read as 1 after it is set to 1 and until VD is input. When VD is input, this bit returns to 0 and so is read as 0. The capture end is determined by the one-frame capture end interrupt (CPE bit). This is similar in data fetch mode. Registers should be set before the VD interrupt of the frame where capture starts next. The new register settings take effect at the next VD input. When registers are modified during capturing, the register settings take effect from the capture operations of the next VD input. If a setting register to which writing during capturing is prohibited is modified during capturing, an interrupt source (IGRW bit in CETCR) is generated. For details on the interrupt source, see the description on CETCR.</li> <li>• Continuous capture When this bit is set to 1 while the CTNCP bit in CAPCR is set to 1, continuous capture starts from the next frame (Figure 50.8). Note that this bit is not cleared to 0 but remains as 1. To stop capturing, clear this bit to 0; capturing stops after the current frame is completed. Continuous capture operations are possible in only image capture mode. The start address of the memory to which the captured data is written to must be set for each frame. 0: Stops capturing 1: Starts capturing</li> </ul>	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
16	CPKIL	Software reset of capturing Write 1 to this bit to perform a software reset of capturing. At a software reset, capturing ends immediately without completing capture operation until the end of a frame. Clear the CE bit to 0 when writing 1 to this bit. Processing of the capture software reset is indicated by this bit being set to 1. When CPKIL = 1, do not start capturing since reset processing is in progress. When restarting capture operations, after referring to the CPTON bit in CSTSR to ensure that the CEU is halted (in the idle state), wait until this bit is cleared to 0. The timing of restarting capture operations is shown in Figure 50.6. When a software reset is generated by this bit, a capture end interrupt (CPE bit in CETCR) may be output immediately after the software reset. However, such kind of interrupt should be ignored. Also, even if the capture end interrupt is not output, the interrupt source (CPE bit) must be cleared before capturing of the next frame. 0: Normal state 1: Software reset of capturing	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CAPSR captures data input to the CEU from an external module.

When both the VD (vertical sync signal) and HD (horizontal sync signal) polarities are high-active, one frame is defined as a period from a VD rising edge to the next VD rising edge, and one line as a period from an HD rising edge to the next HD rising edge. Figure 50.3 shows the timing of one frame (when both the VD and HD polarities are high-active).

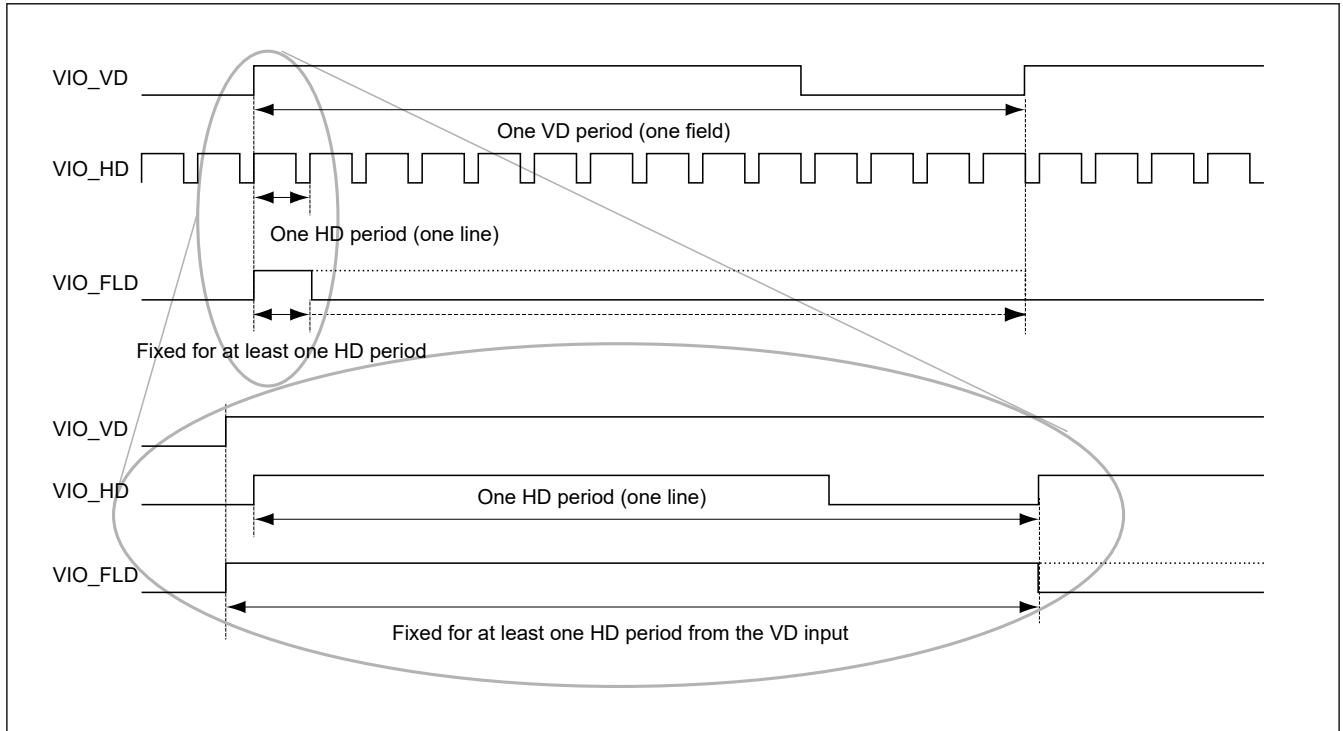


**Figure 50.3** Frame timing

When both the VD and HD polarities are high-active, similar to one frame, one field is defined as follows:

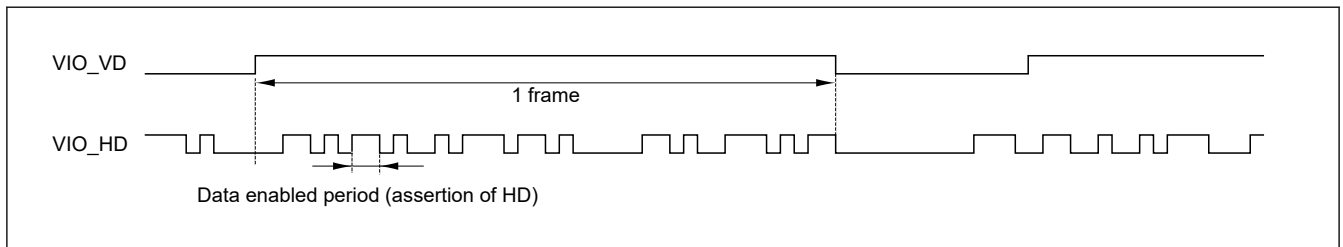
- Period from a VD rising edge to the next VD rising edge
- One line is a period from an HD rising edge to the next HD rising edge

The field identification signal FLD should be fixed for at least 1-HD period from a VD input. [Figure 50.4](#) shows the timing of one field (when both the VD and HD polarities are high-active).



**Figure 50.4 One field timing**

In data enable fetch, one frame is defined as a period from a VD rising edge to the VD falling edge. With the HD as an enable signal (positive polarity), data of a cycle in which the HD is asserted is fetched while the VD is high. [Figure 50.5](#) shows the timing of one frame for data enable fetch.



**Figure 50.5 Frame timing (data enable fetch)**

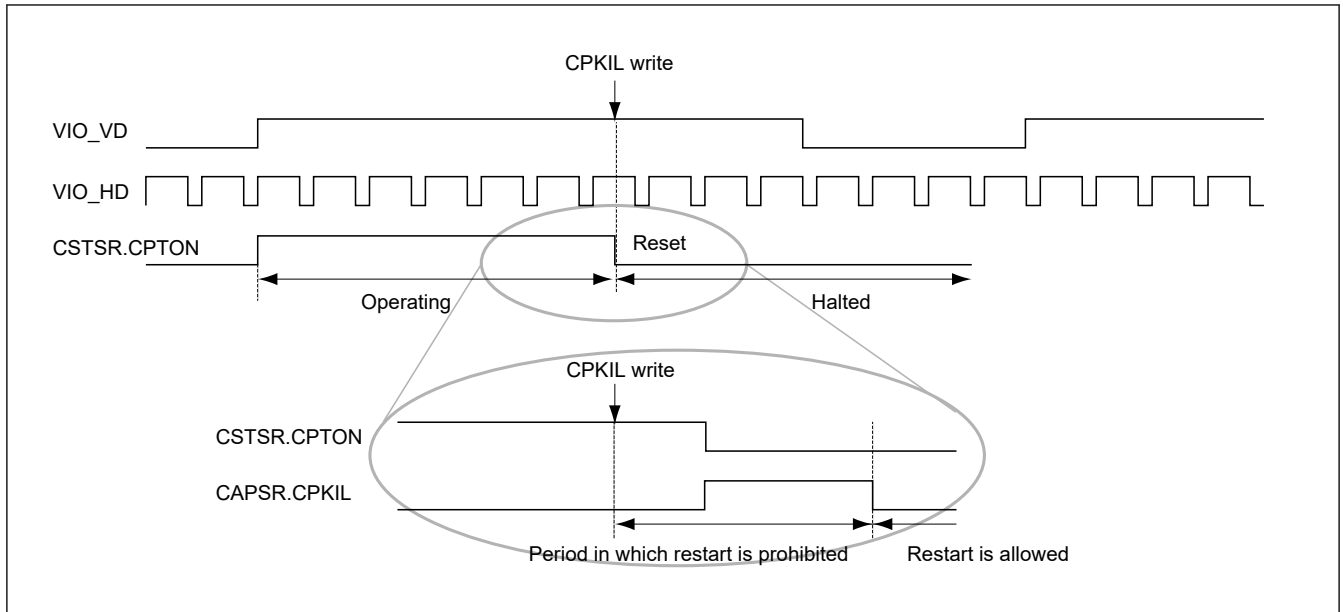


Figure 50.6 Timing of software reset and restart of capturing

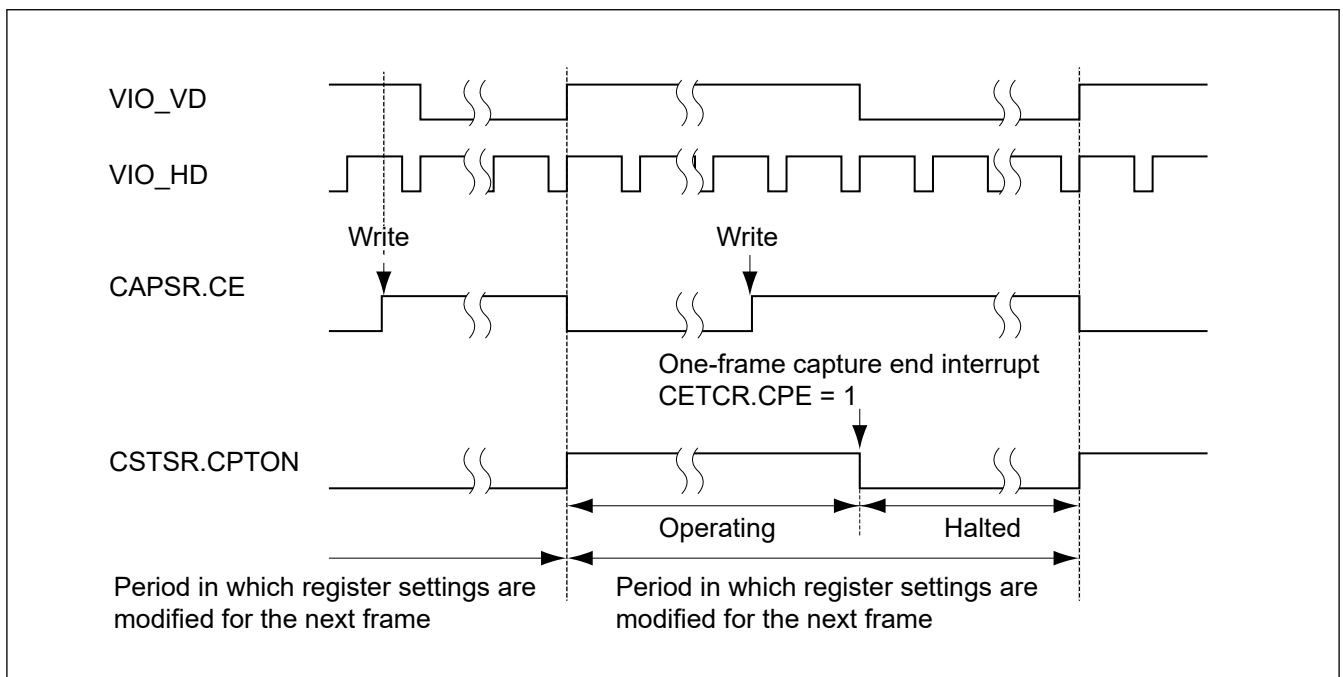


Figure 50.7 Timing of modifying CE bit and register setting in one frame capture

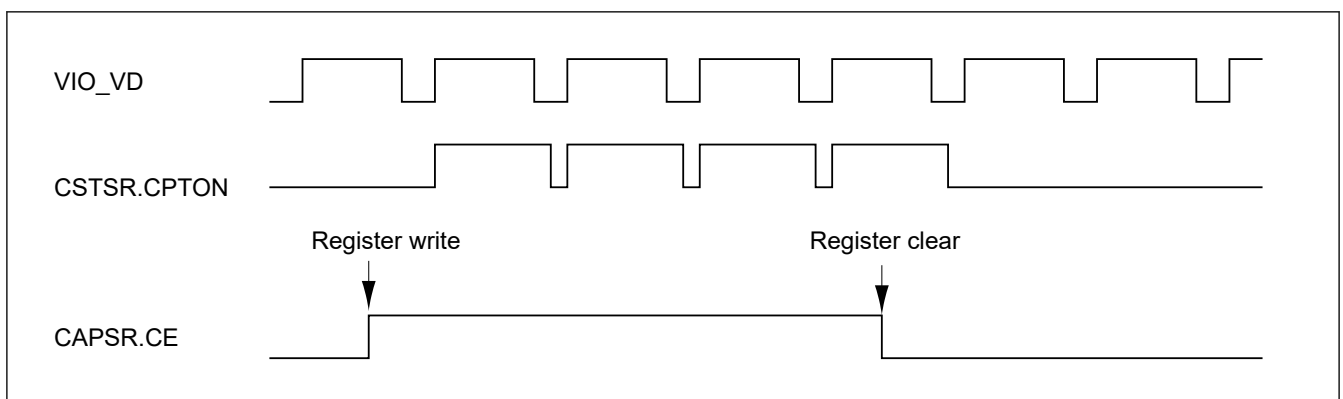


Figure 50.8 Continuous-frame capture

### 50.2.2 CAPCR : Capture Control Register

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	FDRP[7:0]							—	—	MTCM[1:0]		—	—	—	CTNCP	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	CTNCP	Continuous capture When capturing is started with this bit set to 1, capturing continues until the CE bit in CAPSR is cleared to 0 or a software reset is initiated by the CPKIL bit in CAPSR (see <a href="#">Figure 50.8</a> ). Continuous capture must be set before capturing is started. This bit is modified only after 0 is written to the CE bit to stop capturing. If this bit is modified during capturing, correct operation cannot be guaranteed. In data fetch mode, clear this bit to 0.  0: One-frame capture when CAPSR.CE = 1 1: Continuous capture until CAPSR.CE = 0	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
21:20	MTCM[1:0]	Specify the unit for transferring data to a bus bridge module. The efficiency of writing image data can be improved by continuously accessing the addresses. To improve the write efficiency, set these bits to 11. The setting of these bits appear to be unchanged from the outside.  0 0: Image capture: Y data and C data are transferred in 32-byte units Data fetch : Data is transferred in 32-byte units 0 1: Image capture: Y data and C data are transferred in 64-byte units Data fetch : Data is transferred in 64-byte 1 0: Image capture: Y data and C data are transferred in 128-byte units Data fetch : Data is transferred in 128-byte 1 1: Image capture: Y data and C data are transferred in 256-byte units Data fetch : Data is transferred in 256-byte	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	FDRP[7:0]	Set the frame drop interval in continuous-frame capture. When these bits are cleared to 0, frame drop is not performed, and all frames are captured. <a href="#">Figure 50.9</a> shows the value set in these bits and the timing of captured frames. The frame drop interval unit differs according to the capture setting. <a href="#">Table 50.5</a> shows the relationship between the capture setting and frame drop interval unit. The image of the frame drop timing for each capture setting when these bits are set to 2 is shown in <a href="#">Figure 50.10</a> . In both-field capture, capturing is performed continuously for 2-VD periods, regardless of whether the second field is the top field or bottom field. In addition, in both-field capture, the frame drop counter is incremented when the first field has been identified as the top field or bottom field, regardless of whether the second field is the top field or bottom field. When 0 is written to the CE bit in CAPSR, capturing terminates after the current frame has been captured for a capture frame. However, for a drop frame, capturing is forcibly terminated in the CEU so no capture end interrupt (CPE bit in CETCR) is output. While CAPSR.CE = 1, do not change the setting of these bits. Note: Do not change the setting of these bits during continuous capture operations. To change the setting of these bits, stop continuous capture (CAPSR.CE = 0), clear the CTNCP bit (continuous capture) in CAPCR to 0, and then restart continuous capture. Continuous capture is performed during the period of CAPSR.CE = 1 shown in <a href="#">Figure 50.9</a> .	R/W

Note: S-TYPE-3, P-TYPE-3



CAPCR sets continuous-frame capture and the frame drop intervals.

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

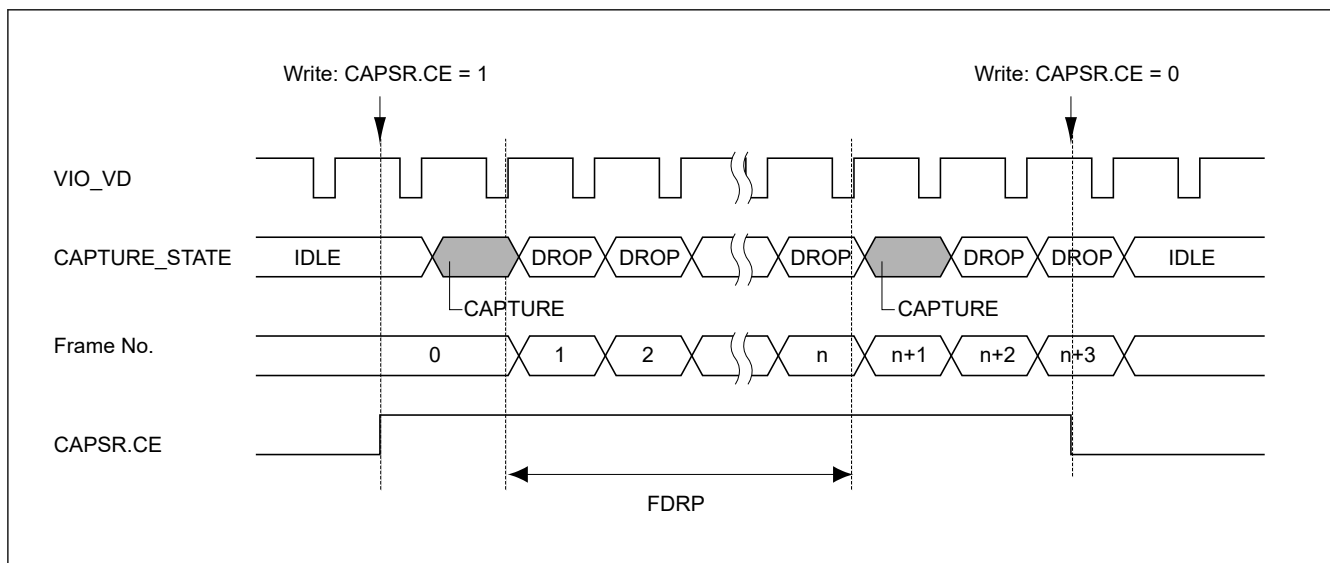


Figure 50.9 Setting of FDRP bits and frame drop timing

Table 50.5 Relationship between capture setting and frame drop interval unit

Input Mode	Captured Image	First Captured Image	Frame Drop Interval Unit	Capture Setting
Progressive	Frame	Frame immediately after capture start	Frame	A
Interlace	Both-field (2-VD capture)	Field immediately after capture start	2 fields (first capture field count)	B
		Top field	2 fields (top-field count)	C
		Bottom field	2 fields (bottom-field count)	D
	One-field (1-VD capture)	Field immediately after capture start	First capture field	F
		Top field	Top field	G
		Bottom field	Bottom field	H
			I	

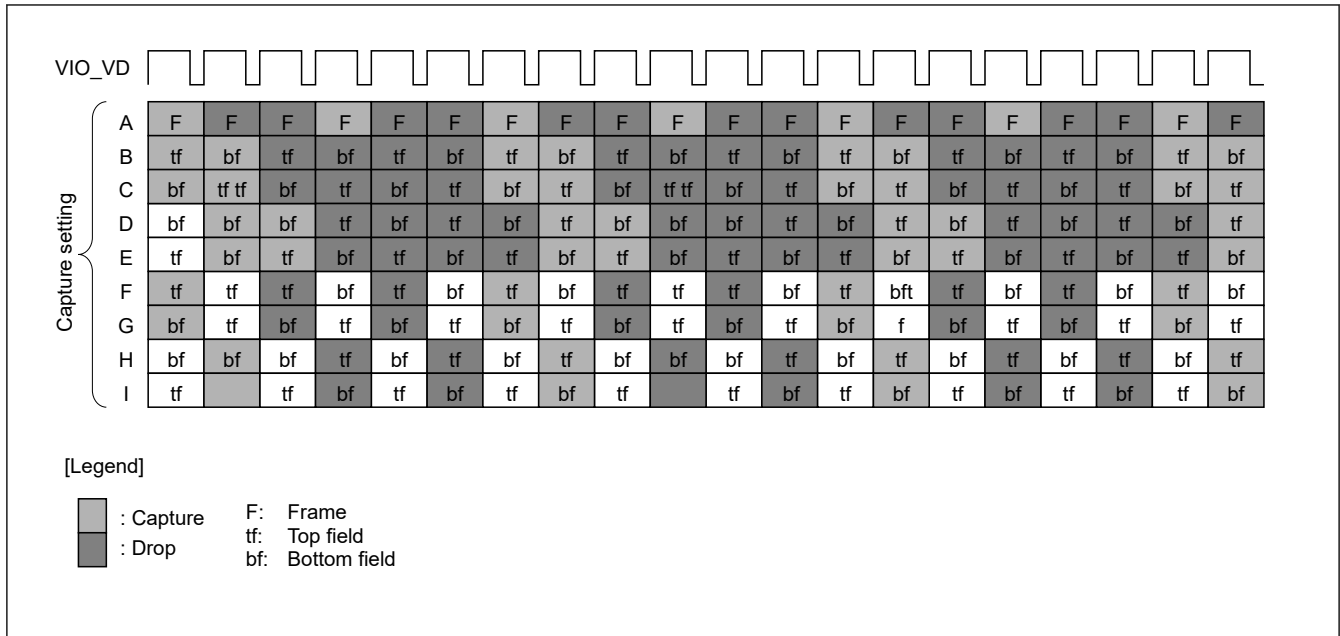


Figure 50.10 Image of frame drop timing for capture settings (FDRP[7:0] = 0x2)

### 50.2.3 CAMCR : Capture Interface Control Register

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	VDSE L	HDSE L	FLDS EL	DSEL	—	—	—	—	—	—	—	FLDP OL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	DTARY[1:0]	—	—	JPG[1:0]	—	—	—	—	VDPO L	HDPO L
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HDPOL	Sets the polarity for detection of the horizontal sync signal (HD) input from an external module. Figure 50.16 shows the relationship between the HD and HD interrupt when high-active is selected. Since an HD interrupt may occur when this bit value is modified, the HD bit in CETCR must always be cleared to 0 when this bit value is changed. In data enable fetch mode, set this bit to 0. 0: Horizontal sync signal (HD) from an external module is detected as high active 1: Horizontal sync signal (HD) from an external module is detected as low active	R/W
1	VDPOL	Sets the polarity for detection of the vertical sync signal (VD) input from an external module. Figure 50.14 and Figure 50.15 show the relationship between the VD, HD and VD interrupt when high-active is selected. Since a VD interrupt may occur when this bit value is modified, the VD bit in CETCR must always be cleared to 0 when this bit value is changed. In data enable fetch mode, this bit is not used and the sense for detection is always active high. 0: Vertical sync signal (VD) from an external module is detected as high active 1: Vertical sync signal (VD) from an external module is detected as low active	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
5:4	JPG[1:0]	These bits select the fetched data type. 0 0: Image capture mode (input data are separated into the luminance component data (Y) and the chrominance component data (CbCr) for output to the memory) 0 1: Data synchronous fetch mode (specified size of input data are output to the specified memory addresses in order of input and in synchronization with the sync signal) 1 0: Data enable fetch mode (input data are fetched with HD as an enable signal and output to the specified addresses in memory in order of input) 1 1: Setting prohibited	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	DTARY[1:0]	Set the input order of the luminance component and chrominance component. The order in which Y and CbCr are input from an external module differs among modules. The CEU supports the input orders shown in Figure 50.12. Set the corresponding value in these bits. In data fetch mode, set these bits to 00. 0 0: Image input data is fetched in the order of Cb <sub>0</sub> , Y <sub>0</sub> , Cr <sub>0</sub> , and Y <sub>1</sub> 0 1: Image input data is fetched in the order of Cr <sub>0</sub> , Y <sub>0</sub> , Cb <sub>0</sub> , and Y <sub>1</sub> 1 0: Image input data is fetched in the order of Y <sub>0</sub> , Cb <sub>0</sub> , Y <sub>1</sub> , and Cr <sub>0</sub> 1 1: Image input data is fetched in the order of Y <sub>0</sub> , Cr <sub>0</sub> , Y <sub>1</sub> , and Cb <sub>0</sub>	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	FLDPOL	Sets the polarity of the field identification signal (FLD) from an external module. 0: When the FLD signal is high-active, the field is detected as the top field and when low-active, the field is detected as the bottom field. 1: When the FLD signal is low-active, the field is detected as the top field and when high-active, the field is detected as the bottom field.	R/W
23:17	—	These bits are read as 0. The write value should be 0.	R/W
24	DSEL	Sets the edge for fetching the image data (D7 to D0) from an external module. 0: D7 to D0 are fetched at the rising edge of the camera clock. 1: D7 to D0 are fetched at the falling edge of the camera clock.	R/W
25	FLDSEL	Sets the edge for capturing FLD from an external module. 0: FLD is captured at the rising edge of the camera clock. 1: FLD is captured at the falling edge of the camera clock.	R/W
26	HDSEL	Sets the edge for capturing HD from an external module. 0: HD is captured at the rising edge of the camera clock. 1: HD is captured at the falling edge of the camera clock.	R/W
27	VDSEL	Sets the edge for capturing VD from an external module. 0: VD is captured at the rising edge of the camera clock. 1: VD is captured at the falling edge of the camera clock.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

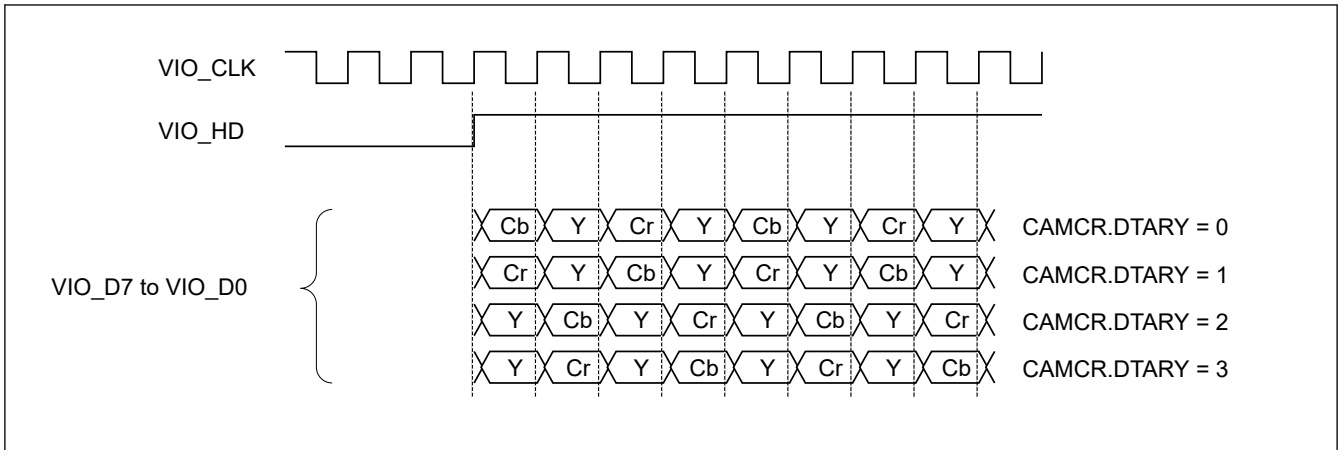
CAMCR sets the capture interface.

The following items are set by CAMCR.

- Selection between the rising edge or falling edge of the camera clock for signal capture or data fetch operation
- Selection between image capture operation or data fetch operation
- Polarities of the vertical and horizontal sync signals
- Input order of image data components (Y, Cb, and Cr) (only for image capture mode)
- Polarity of the field identification signal

CAMCR must be set according to the module connected. In data fetch mode, set the DTARY bits to 0. Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

Note: After changing the setting of this register, do not start capture until at least 10 external input clock cycles have elapsed.



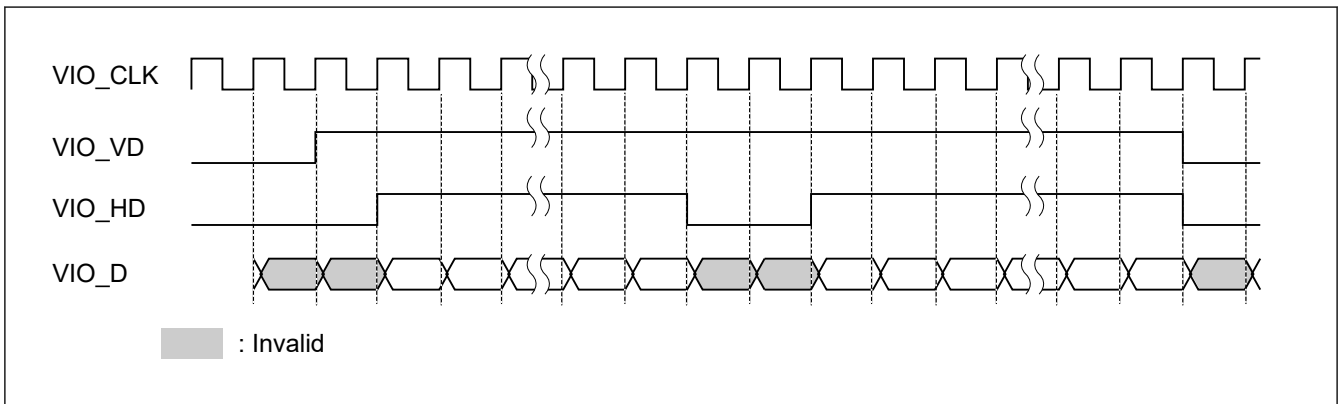
**Figure 50.11 Input order of image data**

The JPG bit in CAMCR selects whether digital image data is fetched or data such as JPEG is fetched. In addition, when data such as JPEG is fetched, select whether the specified amount of data is continuously fetched in synchronization with the sync signal or data is fetched while the horizontal sync signal is enabled.

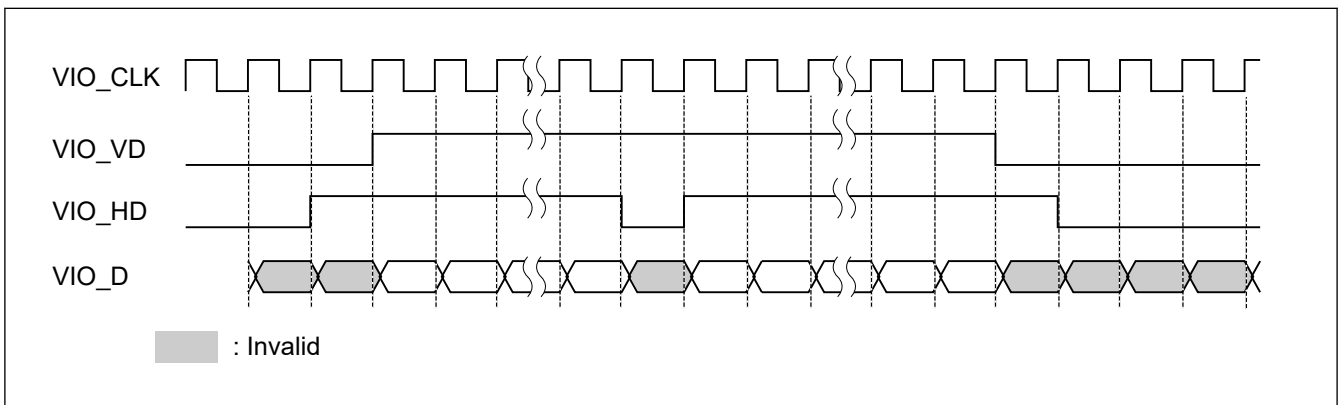
In data enable fetch mode, one frame is defined as a period from the rising edge to the falling edge of the vertical sync signal (VD) for data fetching. The horizontal sync signal (HD) is enabled only when the VD is high and treated as an enable signal. Data input in the cycle in which the HD is asserted (high) is fetched and output to the memory continuously.

This module starts fetching data at the rising edge of the VD and stops fetching data at the falling edge of the VD in data enable fetch mode. Thus, if the VD remains high and does not go low, end processing does not start. In addition, if the VD remains high and the HD also remains asserted, data continues to be fetched.

Figure 50.12 and Figure 50.13 show the interface timing in data enable fetch mode.



**Figure 50.12 Data enable fetch timing (HD asserted (high) while VD is high)**



**Figure 50.13 Data enable fetch timing (HD asserted (high) when VD is not high)**

In data enable fetch mode, this module generates a VD interrupt in response to detection of the active level of VD. In image capture mode and data synchronous fetch mode, this module generates a VD interrupt in response to the first detection of the active level of HD following detection of the active level of VD. Note that, when VD and HD are asserted and detected at the same time, this module generates a VD interrupt at that time.

Figure 50.14 to Figure 50.16 show the relationships between the VIO\_VD signal and the VD interrupt, the VIO\_VD and VIO\_HD signals and the VD interrupt, and the VIO\_HD signal and the HD interrupt.

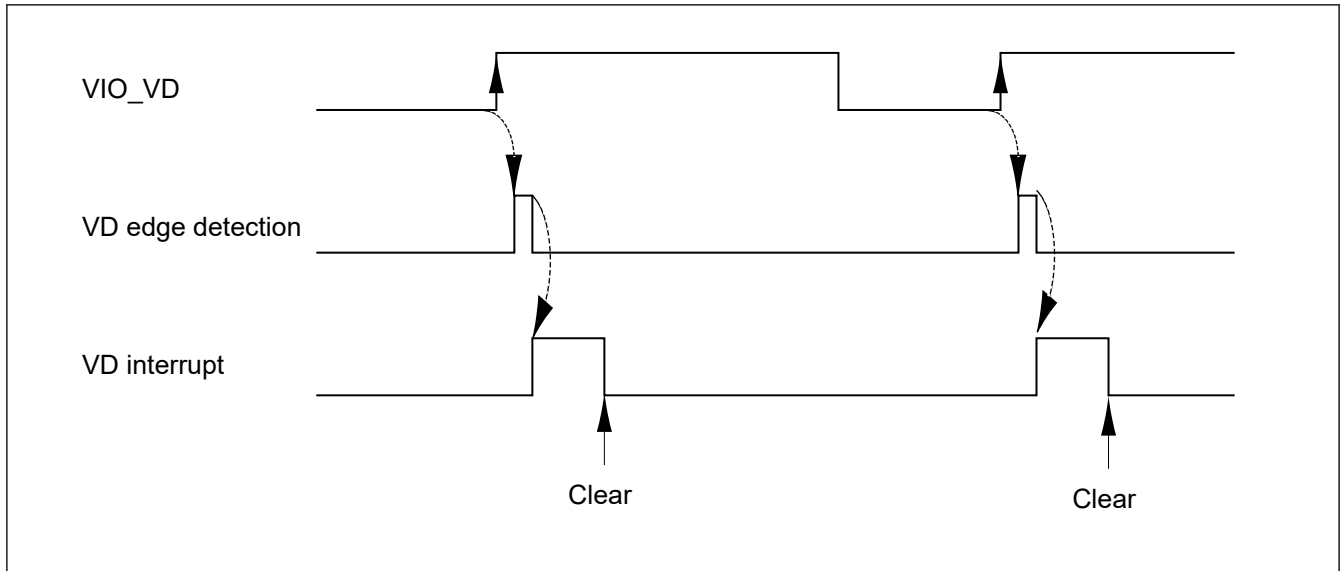


Figure 50.14 Relationship between VIO\_VD and VD interrupt when VD is high-active (in data enable fetch mode)

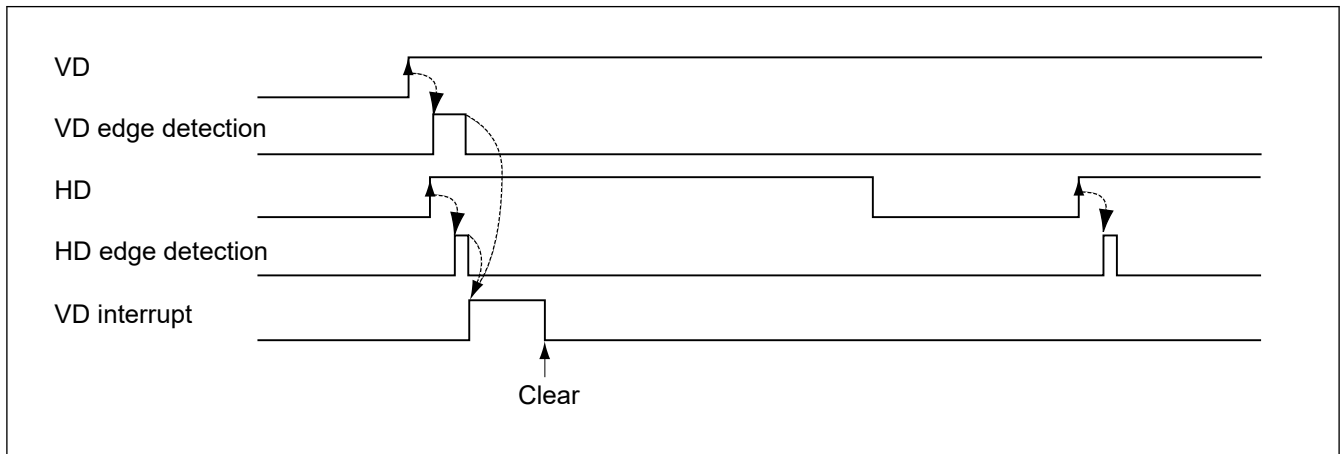


Figure 50.15 Relationship between the VIO\_VD and VIO\_HD signals and the VD Interrupt when VD and HD are high-active (in Image capture mode or data synchronous fetch mode)

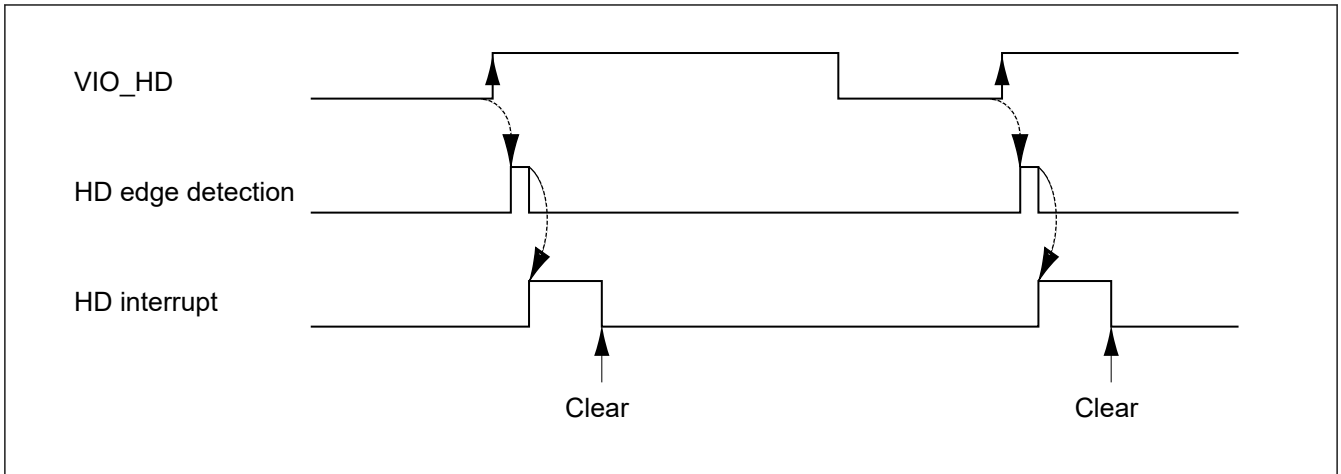
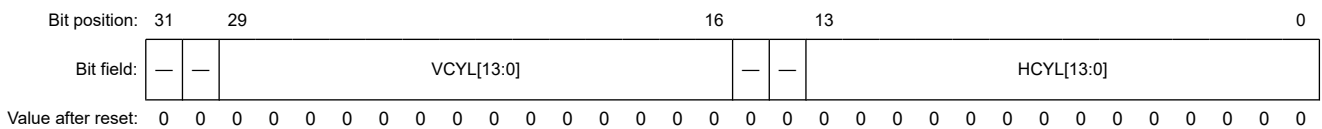


Figure 50.16 Relationship between VIO\_HD and HD interrupt when HD is high-active

### 50.2.4 CMCYR : Capture Interface Cycle Register

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x000C



Bit	Symbol	Function	R/W
13:0	HCYL[13:0]	Horizontal Cycle Count of External Module These bits set the number of HD cycles of an external module. The interrupt source bit IGHS in CETCR is set to 1 when the actual number of HD cycles input from the external module differs from this setting. Set these bits for detecting an illegal HD. When these bits are all cleared to 0, the interrupt source bit IGHS in CETCR is not set to 1. Though the interrupt source bit IGHS in CETCR may be set to 1 after the HDPOL bit (HD polarity) in CAMCR is changed, this interrupt should be ignored.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
29:16	VCYL[13:0]	Vertical HD Count of External Module These bits set the number of VD cycles of an external module with the number of HD inputs. The interrupt source bit IGVS in CETCR is set to 1 when the actual number of VD cycles input from the external module differs from this setting. Set these bits for detecting an illegal VD. When these bits are all cleared to 0, the interrupt source bit IGVS in CETCR is not set to 1. Though the interrupt source bit IGVS in CETCR may be set to 1 after the VDPOL bit (VD polarity) in CAMCR is changed, this interrupt should be ignored.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CMCYR is used to detect an illegal VD and an illegal HD. For HD, the number of cycles from a rising edge of HD to the next rising edge is set (falling edges when low-active is selected for HD). For VD, the number of HD inputs from a rising edge of VD to the next rising edge is set (falling edges when low-active is selected for VD).

Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

Set 0 in all bits of this register, during data enable fetch mode.

Note: After changing the setting of this register, do not start capture until at least 10 external input clock cycles have elapsed.



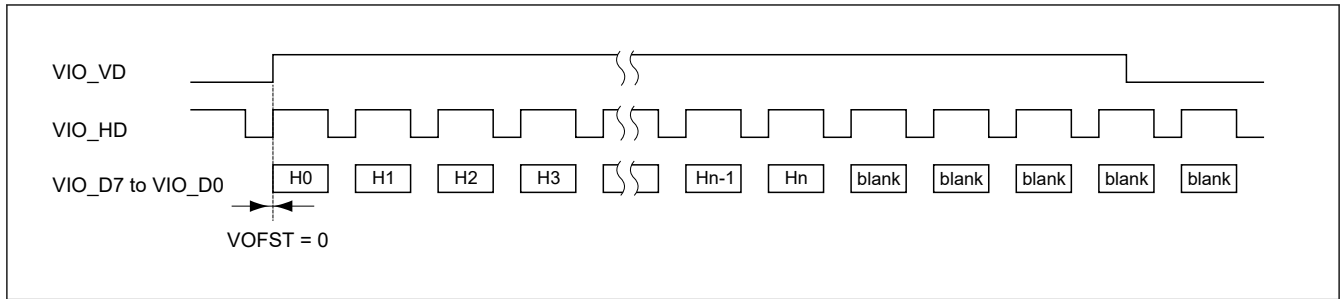


Figure 50.18 Timing when VD is data enable signal

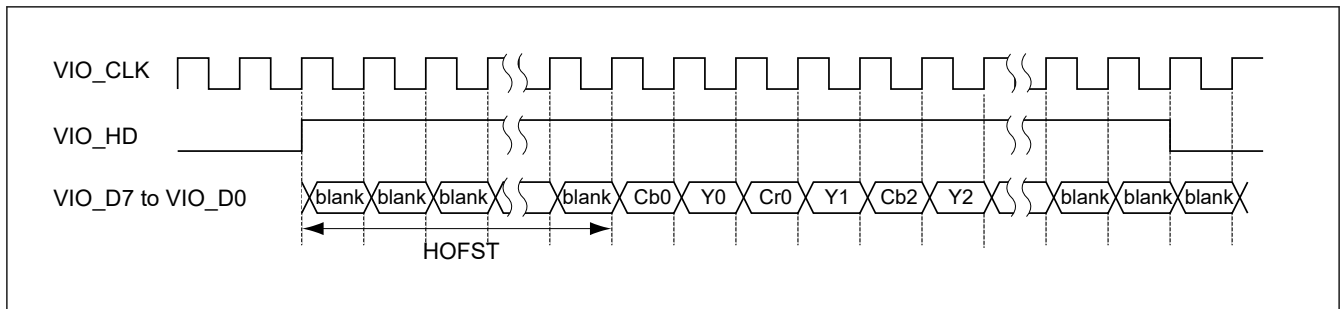


Figure 50.19 Horizontal offset

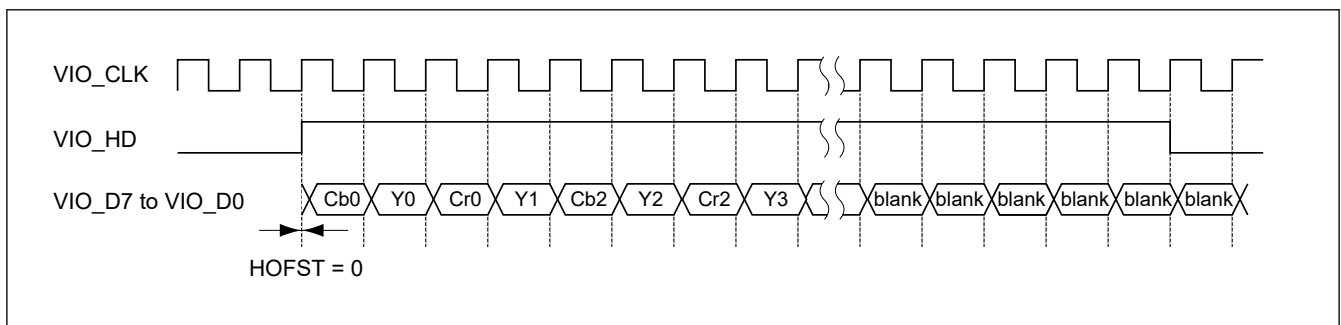
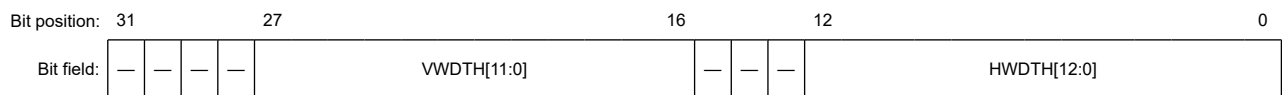


Figure 50.20 Timing when HD is data enable signal

### 50.2.6 CAPWR, CAPWR\_x : Capture Interface Width Register (x = B, M)

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0014 (CAPWR)  
 0x1014 (CAPWR\_B)  
 0x2014 (CAPWR\_M)



Value after reset: 0



Bit	Symbol	Function	R/W
12:0	HWDTH[12:0]	Specify the horizontal capture period. Specify the number of cycles to be captured from the location specified by the HOFST bits. Figure 50.22 shows the timing when the horizontal blanking period is 0. The CEU captures for only the number of cycles specified by these bits in the horizontal direction. Make a similar setting for data synchronous fetch. The maximum value to be set is as follows: Image capture (16-cycle units) : 5,120 cycles (2,560 pixels) Data synchronous fetch (8-cycle units) : 2,560 cycles (2,560 bytes) The lower 2 bits should be written 0. Note: In data synchronous fetch mode, set CFSZR and CDWDR according to the values set in this register. For details, see the descriptions on CFSZR and CDWDR.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
27:16	VWDTH[11:0]	Specify the vertical capture period (4-HD units). Specify the number of lines (HD count) to be captured from the location specified by the VOFST bits in CAMOR. Figure 50.21 shows the timing when the vertical blanking period is 0. The CEU captures only the number of lines (HD count) specified by these bits in the vertical direction. Make a setting in the same way to obtain data synchronization. The maximum value to be set is 1,920 HD (5 megapixels). The value of writing are ignored on the lower 2 bits, so write 0 to the lower 2 bits.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

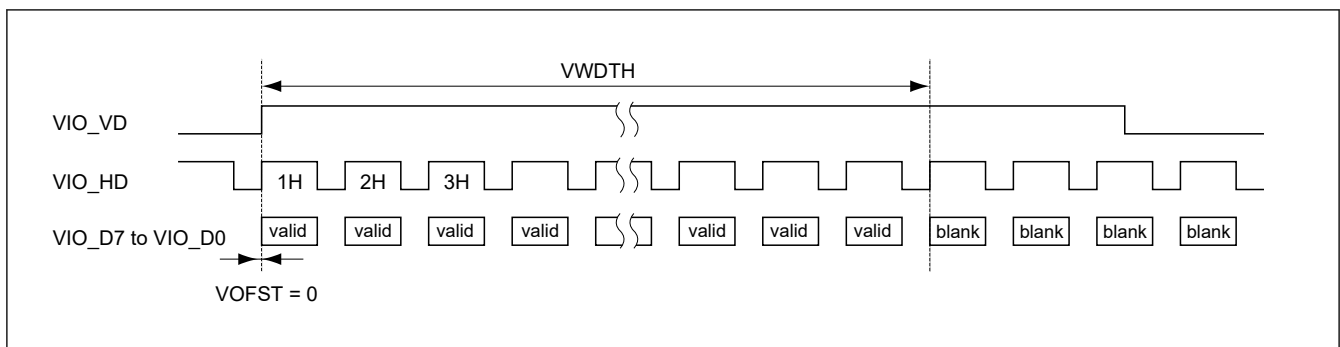
CAPWR sets the fetch (capture) cycle width when capturing images.

The cycle width unit differs according to the interface and the data type to be captured. For each setting unit, see Table 50.6.

This register is not used, during data enable fetch mode.

**Table 50.6 Unit for setting fetch (capture) cycle width**

Vertical Direction		Horizontal Direction	
Image Capture	Data Synchronous Fetch	Image Capture	Data Synchronous Fetch
4 HD	4 HD	16 cycles	8 cycles



**Figure 50.21 Vertical capture timing**

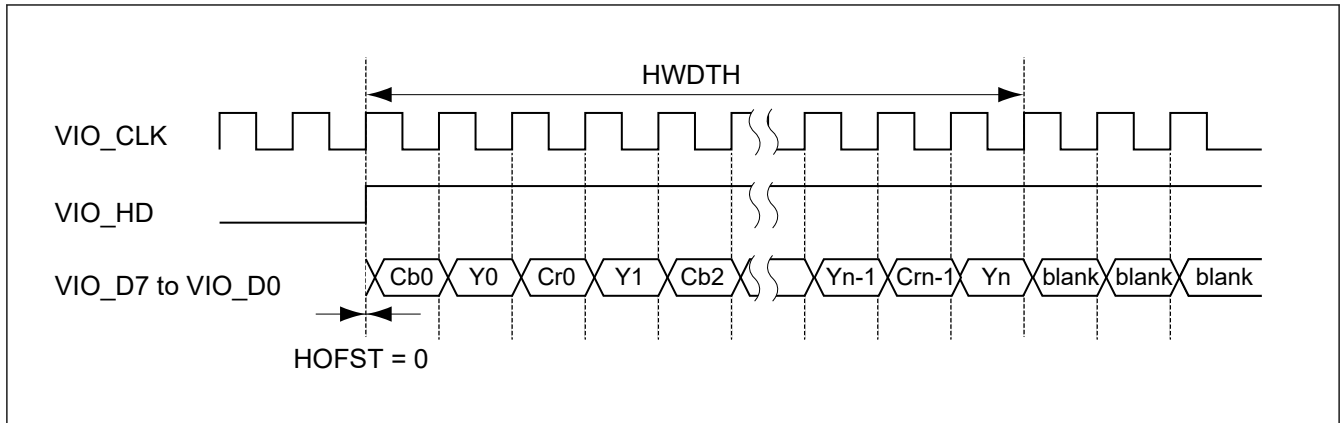


Figure 50.22 Horizontal capture timing

### 50.2.7 CAIFR : Capture Interface Input Format Register

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	IFS	—	—	—	CIM	—	—	—	—	FCI[1:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	FCI[1:0]	Set the timing to start capturing. The timing to start capturing is set by specifying the image to be captured first. Set these bits to 00 when the input mode is progressive. 0 0: Capture starts from the VD input immediately after the CEU activation regardless of it being a top or bottom field 0 1: After the CEU activation, input of a top-field image is waited, and then capture starts from the top field 1 0: After the CEU activation, input of a bottom-field image is waited, and then capture starts from the bottom field 1 1: Setting prohibited	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	CIM	Sets the images to be captured. Clear this bit to 0 when the input mode for image capture is progressive (frame image) or when the input mode for image capture is interlace for continuous capture of both the top and bottom fields. Set this bit to 1 when the input mode for image capture is interlace for capture of only a one-field image. 0: Capture of frame image (1 VD) or both-field image (2 VD) 1: Capture of one-field image (1 VD)	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	IFS	Sets the input mode for capturing images. 0: Progressive 1: Interlace	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CAIFR sets the input mode (progressive or interlace) for capturing images, the images to be captured (frame, both-field, or one-field), the image from which capturing starts (top field or bottom field), etc. CAIFR is not used in data fetch mode.

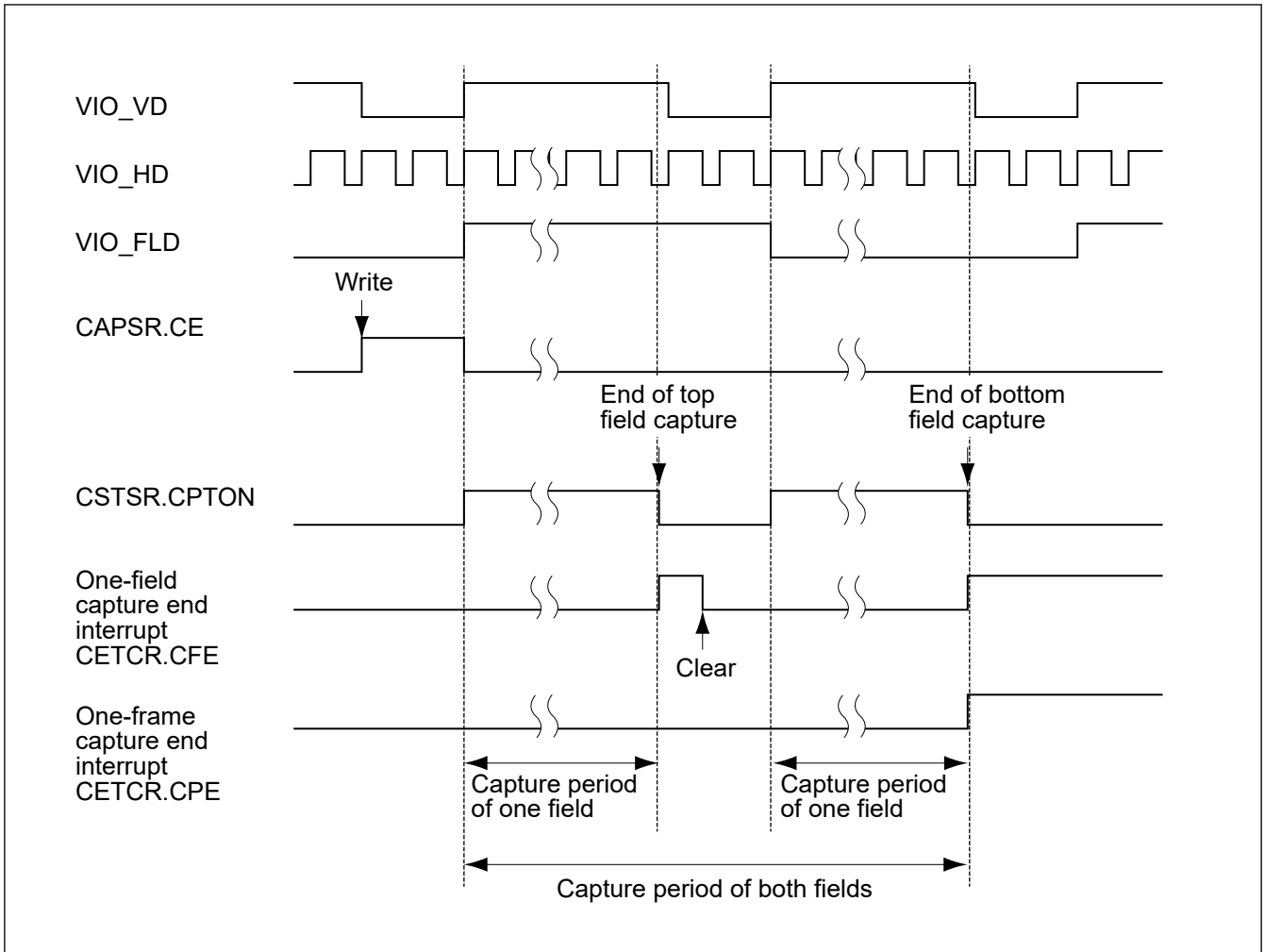
Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

The items set by CAIFR are listed in [Table 50.7](#).

**Table 50.7 CAIFR Setting Items**

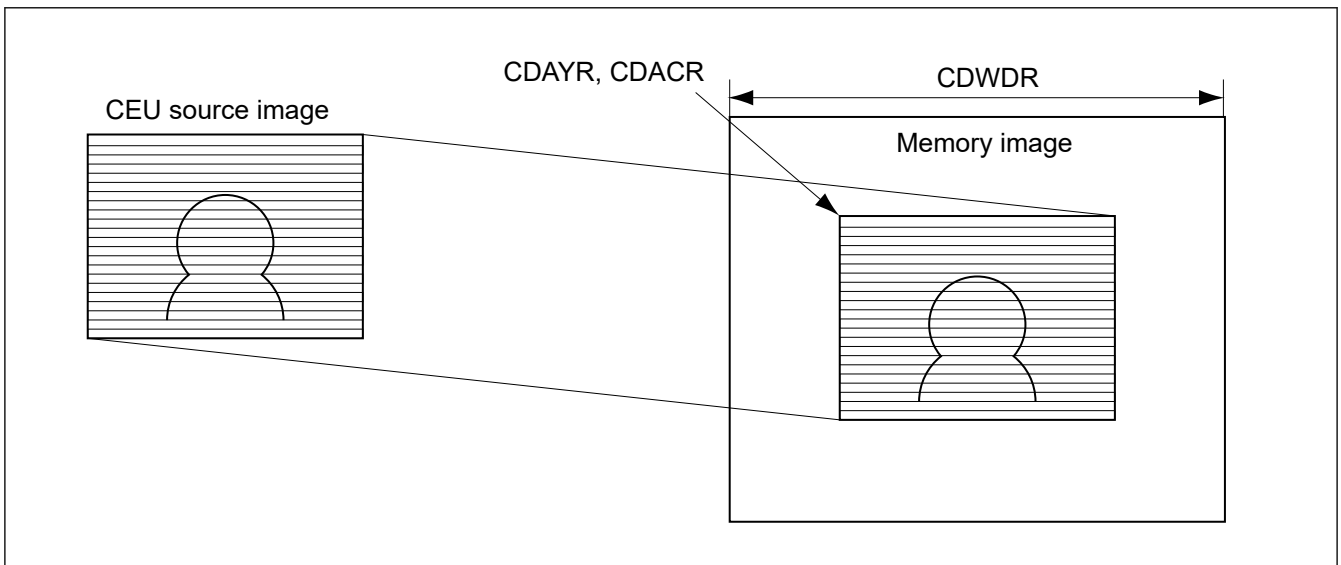
Input Mode	IFS Bit	Captured Image	CIM Bit	Image to Start Capture	FCI Bits
Progressive	0	Frame	0	Frame immediately after activation	00
Interlace	1	Both-field (2-VD capture)	0	Field immediately after activation	00
				Top field	01
				Bottom field	10
				Setting prohibited	11
		One-field (1-VD capture)	1	Field immediately after activation	00
				Top field	01
				Bottom field	10
				Setting prohibited	11

In frame image capture and one-field image capture, a one-frame capture end interrupt occurs when capture for 1 VD finishes. In both-field image capture, a one-field capture end interrupt occurs when capture for 1 VD finishes and a oneframe capture end interrupt occurs when capture for 2 VD finishes. At this time, a one-field capture end interrupt occurs simultaneously with a one-frame capture end interrupt. [Figure 50.23](#) shows the timing of a one-frame capture end interrupt and one-field capture end interrupt in both-field image capture.

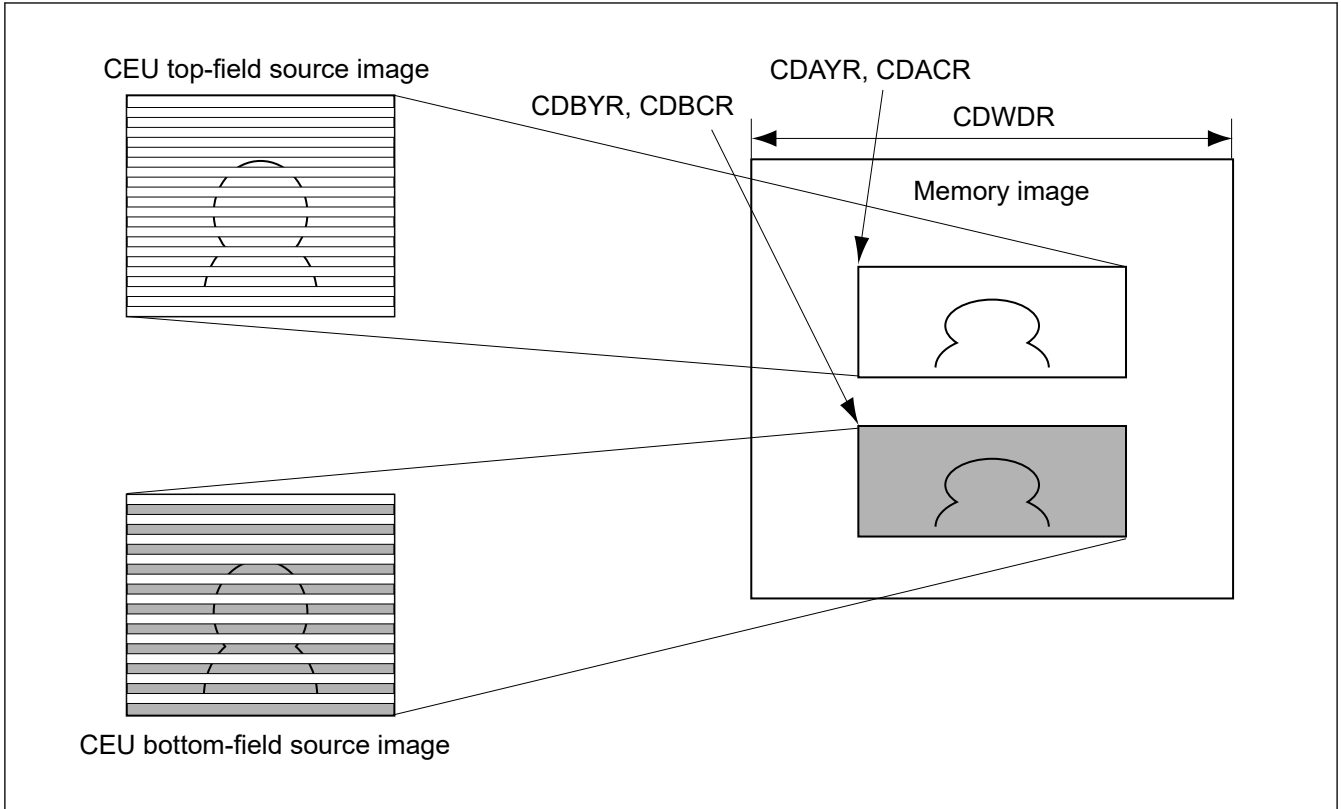


**Figure 50.23 One-frame capture end interrupt and one-field capture end interrupt in both-field image capture**

A captured frame image or captured one-field image is stored in the memory from the addresses set in CDAYR and CDACR (Figure 50.24). Captured both-field images are stored in different memory areas depending on whether it is a top-field or bottom-field image. A top-field image is stored in the memory from the addresses set in CDAYR and CDACR whereas a bottom-field image is stored in the memory from the addresses set in CDBYR and CDBCR (Figure 50.25).

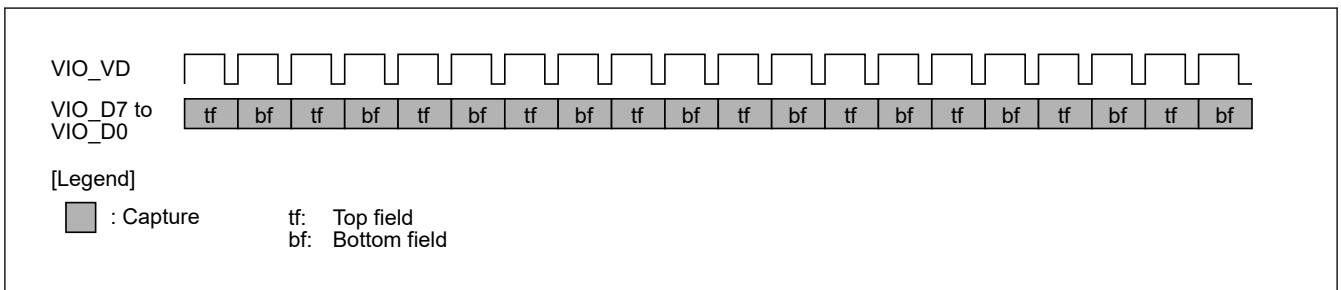


**Figure 50.24 Image of storing captured frame image or captured one-field image in memory**

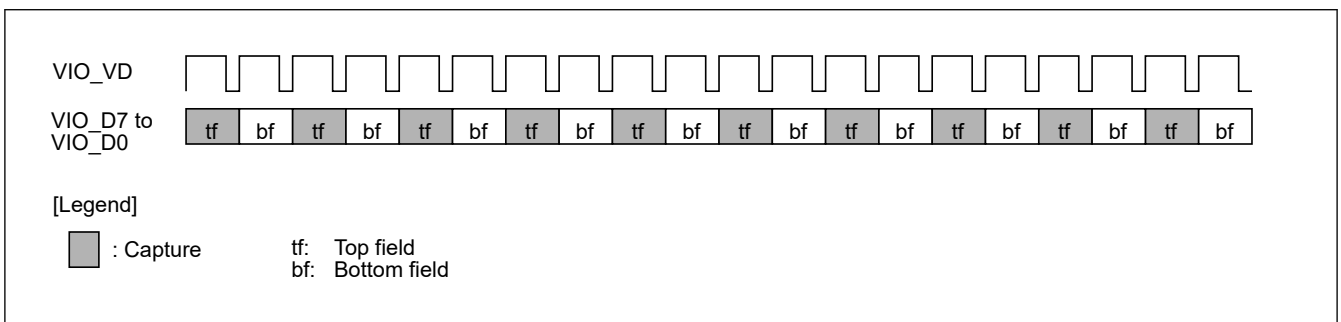


**Figure 50.25 Image of storing captured both-field images in memory**

If the FCI bits are set to 00b for continuous capture in interlace input mode, images are continuously captured for 2 VD with the first captured field as the reference in both-field image capture (Figure 50.26). In one-field image capture, only the first captured field is continuously captured for 1 VD (Figure 50.27).



**Figure 50.26 Continuous both-field capture in interlace mode(Image Immediately after Activation is Top Field (FCI[1:0] = 00b))**



**Figure 50.27 Continuous one-field capture in interlace mode(Image Immediately after Activation is Top Field (FCI[1:0] = 00b))**

### (1) Storage of Interlace Input as Frame Image

The CEU can store an interlace source image in the memory as a frame image. To store an interlace source image as a frame image, make the following register settings:

Input mode: Interlace (IFS = 1)

Capture image: Both-field (CIM = 0)

Image to start capture: Any setting other than the prohibited setting (FCI[1:0] = as desired)

Figure 50.28 shows a memory image of capturing both fields of an interlace input and storing it as a frame image in the memory. Set the start addresses of the memory destination for the captured top-field image in CDAYR and CDACR, and the start addresses of the memory destination for the captured bottom-field image in CDBYR and CDBCR. When storing an interlace image as a frame image in the memory, set the horizontal image size of the memory area in CDWDR with the top-field image and bottom-field image placed next to each other as shown in Figure 50.28. In addition, set the number of captured lines of the field image in the VWDTH bits in CAPWR.

A memory image of folding the horizontal image size of the memory area in Figure 50.28 at CDWDR/2 is shown in Figure 50.29. Setting the registers to form the image in Figure 50.28 enables an interlace image to be stored as a frame image in the memory as shown in Figure 50.29.

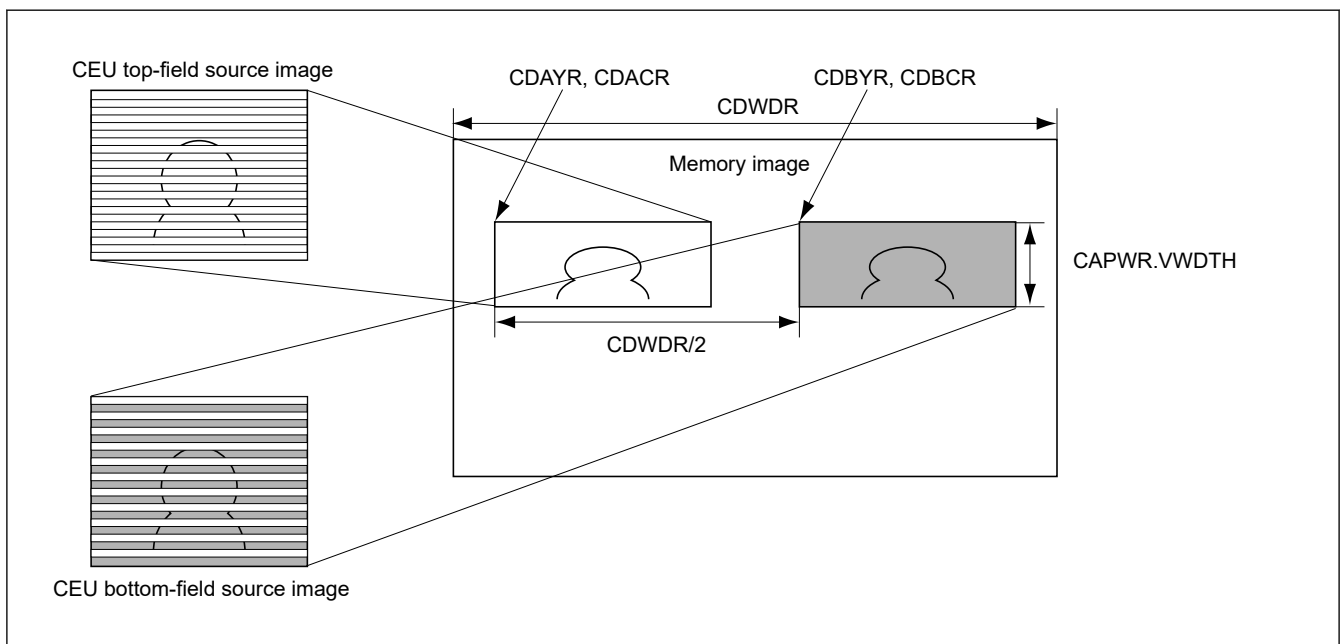


Figure 50.28 Image of storing captured both-fields of interlace input in memory

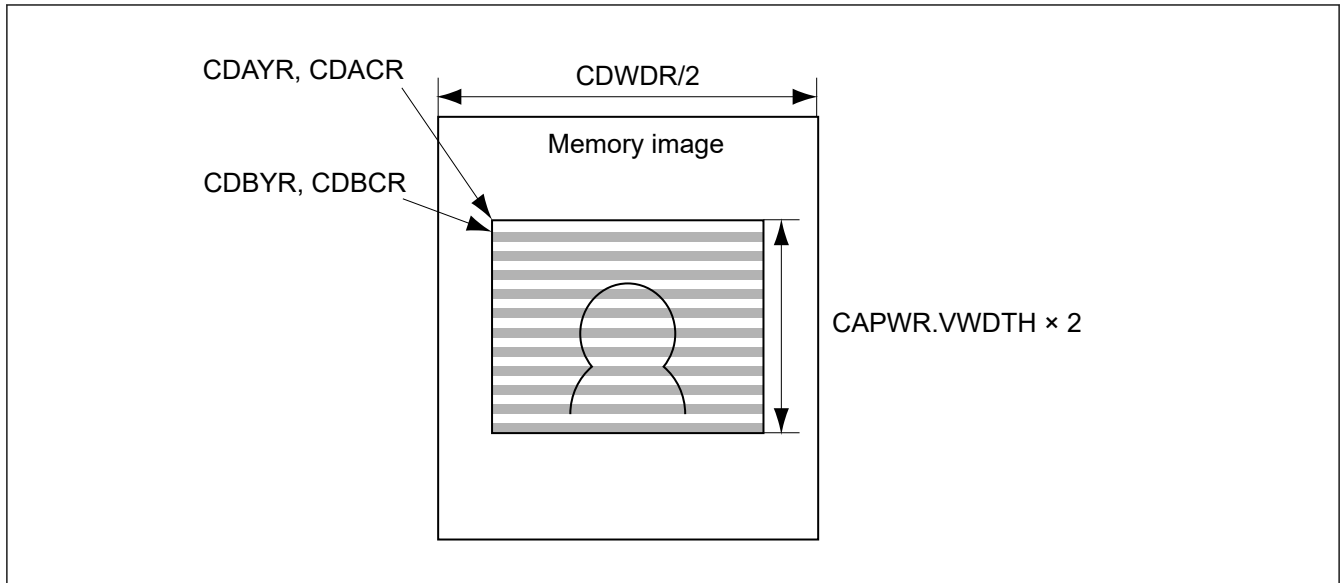


Figure 50.29 Image of storing Interlace input as frame image in memory

### 50.2.8 CRCNTR : CEU Register Control Register

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RVS	—	—	RS	RC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RC	Specifies switching of the register plane used by the CEU in synchronization with VD. If the register plane is not switched, the register plane specified by the RS bit is used. 0: Uses the specified register plane in synchronization with VD 1: Switches the register plane in synchronization with VD	R/W
1	RS	Specifies which register plane is used by the CEU in synchronization with VD. The setting of this bit is valid only when RC = 0. 0: Uses plane A of the register 1: Uses plane B of the register	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	RVS	Sets the timing to switch the register plane in both-field capture. The setting of this bit is valid only when RC = 1 in both-field capture. 0: Switches the register plane every 2 VD 1: Switches the register plane every 1 VD	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CRCNTR controls switching of the planes of registers with a 2-plane configuration.

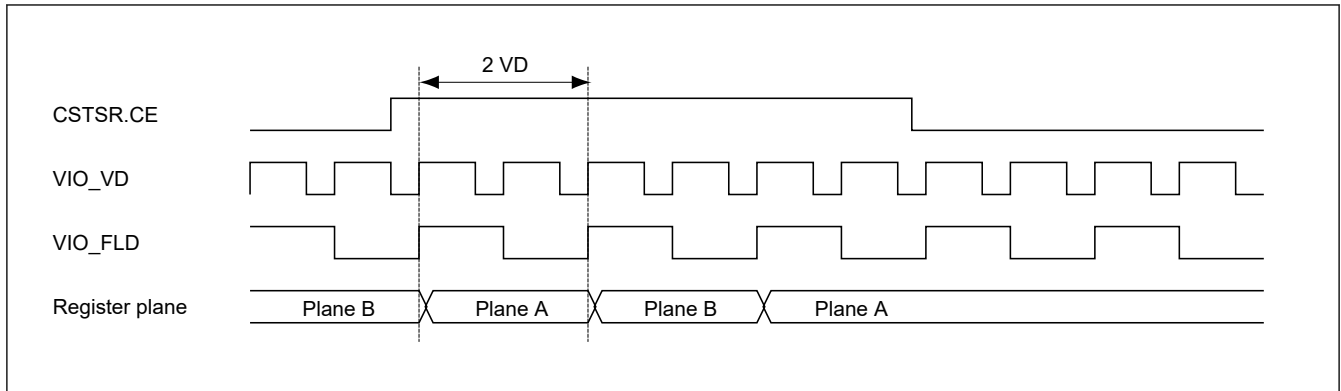


Figure 50.30 Timing for register plane switching when RVS = 0

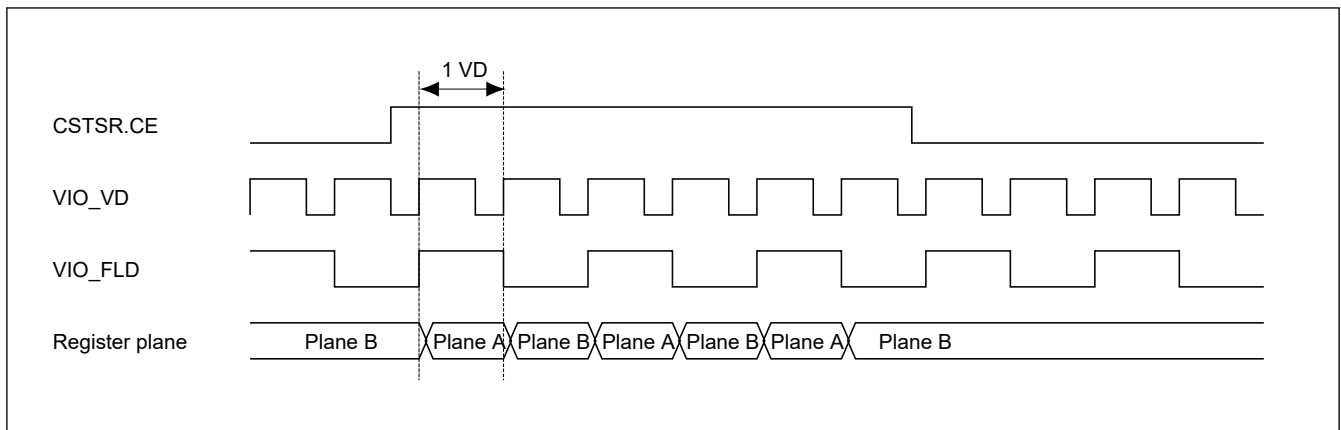


Figure 50.31 Timing for register plane switching when RVS = 1

### 50.2.9 CRCMPR : CEU Register Forcible Control Register

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x002C

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

Bit	Symbol	Function	R/W
0	RA	Indicates the register plane currently specified. This register value automatically changes in synchronization with VD for starting capture. To start capture with plane A of the register when a setting to switch the register plane in synchronization with VD has been made (CRCNTR.RC = 1), specify plane B of the register using this bit. 0: Specifies plane A of the register 1: Specifies plane B of the register	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CRCMPR forcibly controls switching of the planes of registers with a 2-plane configuration. Setting this register enables direct control of register plane switching.

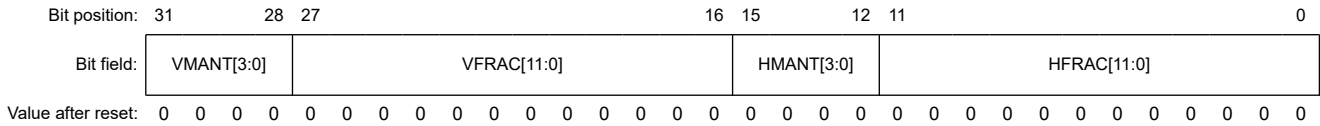


Do not modify this register during operation. If this register is modified during operation, correct operation cannot be guaranteed. In addition, the IGRW bit (interrupt source) in CETCR is set to 1.

### 50.2.10 CFLCR, CFLCR\_x : Capture Filter Control Register (x = B, M)

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0030 (CFLCR)  
 0x1030 (CFLCR\_B)  
 0x2030 (CFLCR\_M)



Bit	Symbol	Function	R/W
11:0	HFRAC[11:0]	Fraction Part of Horizontal Scale-Down Factor The specifiable range is 0x000 to 0xFF8. The fraction of the scale-down factor that cannot be set with only the HMANT bits must be specified with these bits. The value of writing are ignored on the lower 3 bits, so write 0 to the lower 3 bits.	R/W
15:12	HMANT[3:0]	Mantissa Part of Horizontal Scale-Down Factor The specifiable range is 0x0 to 0xF. When 0x0 is set for the HMANT bits and 0x000 is set for the HFRAC bits, the scale-down filter is not used.	R/W
27:16	VFRAC[11:0]	Fraction Part of Vertical Scale-Down Factor The specifiable range is 0x000 to 0xFF8. The fraction of the scale-down factor that cannot be set with only the VMANT bits must be set with these bits. The value of writing are ignored on the lower 3 bits, so write 0 to the lower 3 bits.	R/W
31:28	VMANT[3:0]	Mantissa Part of Vertical Scale-Down Factor The specifiable range is 0x0 to 0xF. When 0x0 is set for the VMANT bits and 0x000 is set for the VFRAC bits, the scale-down filter is not used.	R/W

Note: S-TYPE-3, P-TYPE-3

CFLCR sets the scale-down factor for the filter to scale images down.

The CEU has an image scale-down filter which can be used to scale down the captured images before storing them in the memory. Set CFLCR to 0 when not performing scale-down (same size output). If a value other than 0 is set in CFLCR, scale-down is performed. In data fetch mode, set CFLCR to 0.

When handling an interlace source image as a frame image, set CFLCR to 0 not to use the filter.

An image scale-down filter is installed in the CEU, and the captured images can be scaled down and stored in the memory.

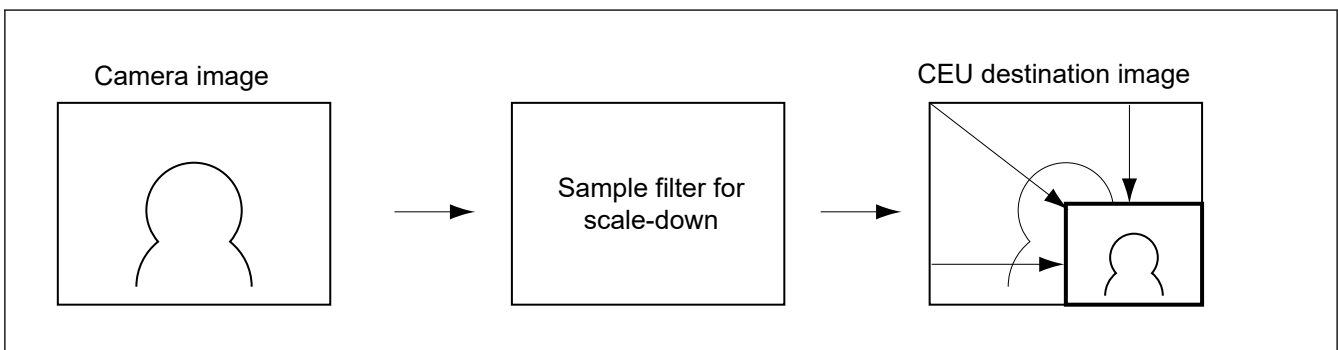


Figure 50.32 Scale-down of captured image

The formulas for obtaining the MANT (VMANT or HMANT) and FRAC (VFRAC or HFRAC) values from the input pixel count and output pixel count of the filter are shown below. Set the MANT and FRAC bits in order to obtain the desired output pixel count from the number of pixels input to the CEU.

First, calculate preliminary MANT and FRAC values. The parameters needed for calculation are defined as follows:

$$\alpha = \text{MANT} \times 4096 + \text{FRAC} \quad \dots\text{Formula 1}$$

$$\text{SCL (scaling factor)} = \frac{4096}{\alpha} \quad \dots\text{Formula 2}$$

Assuming an operator  $\lfloor x \rfloor$  which discards fractions of an integer  $x$ , the MANT and FRAC values can be temporarily set as follows, according to formula 1 and formula 2.

$$\text{MANT} = \left\lfloor \frac{1}{\text{SCL}} \right\rfloor, \text{FRAC} = \left\lfloor 512 \times \left( \frac{1}{\text{SCL}} - \text{MANT} \right) \right\rfloor \times 8$$

Here, the scaled-down filter output size ( $\text{SIZE}_D$ ) can be calculated using the input image size  $S_{in}$  (half of the CAPWR setting) in the following formula.

$$\text{SIZE}_D = \left\lfloor 1 + \left( \left\lfloor \frac{1}{2} + \frac{S_{in}-1}{\text{MANT}_{pre}} \right\rfloor - 1 \right) \times \frac{\text{MANT}_{pre} \times 4096}{\alpha} \right\rfloor \quad \dots\text{Formula 3}$$

$$\left[ \begin{array}{l} \text{MANT}_{pre} = 1 \rightarrow (0 \leq \text{MANT} < 2) \\ \text{MANT}_{pre} = 2 \rightarrow (2 \leq \text{MANT} < 4) \\ \text{MANT}_{pre} = 4 \rightarrow (4 \leq \text{MANT} < 8) \\ \text{MANT}_{pre} = 8 \rightarrow (8 \leq \text{MANT}) \end{array} \right]$$

The number of output pixels can be obtained by substituting the temporarily calculated MANT, FRAC, and input image size into these formulas. If the calculated number of output pixels is smaller than the number of output pixels used to obtain the preliminary MANT and FRAC values, recalculate with a smaller FRAC ( $\alpha$ ) value, and set the MANT and FRAC values in this register so that a pixel value greater than the desired number of output pixels can be obtained.

Example: Scale down 640 pixels to 480 pixels

$\text{SCL} = 480/640 = 3/4$ , and the preliminary settings of  $\text{MANT} = 1$ ,  $\text{MANT}_{pre} = 1$ , and  $\text{FRAC} = 0x550$  are made. Substituting these in the following formula results in an output pixel count of 479.

$$\text{SIZE}_D = \left\lfloor 1 + \left( \left\lfloor \frac{1}{2} + \frac{S_{in}-1}{\text{MANT}_{pre}} \right\rfloor - 1 \right) \times \frac{\text{MANT}_{pre} \times 4096}{\alpha} \right\rfloor \quad \dots\text{Formula 3}$$

Since this output pixel count is smaller than the desired output pixel count of 480, the formula is recalculated with a FRAC value of  $0x548$ , a value eight less than the previous time. The obtained result of output pixel count = 480 is equal to the desired output pixel count of 480, so this register is set as  $\text{MANT} = 1$  and  $\text{FRAC} = 0x548$ .

**Table 50.8 Setting examples for each scale-down filter factor**

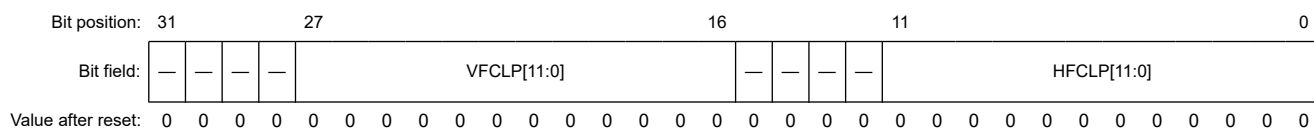
Scale-down factor	FRAC		MANT	Input pixel count	Output pixel count	Clipping size (CFSZR)
	Decimal	Hexadecimal				
7/8	576	0x240	1	640	560	560
3/4	1352	0x548	1	640	480	480
5/8	2448	0x990	1	640	400	400
1/2	0	0x0	2	640	320	320
3/8	2728	0xAA8	2	640	240	240
1/3	0.0	0x0	3	640	213	212
1/4	0.0	0x0	4	640	160	160
1/5	0.0	0x0	5	640	128	128
1/6	0.0	0x0	6	640	107	104
1/7	0.0	0x0	7	640	91	88
1/8	0.0	0x0	8	640	80	80
1/16	4088	0xFF8	15	640	40	40

Note: This scale-down filter uses a VGA-size line memory for scale-down. Therefore, when an image larger than the VGA size is input for scale-down, settings must be made so that the output image size is equal to or larger than the SubQCIF size and equal to or smaller than the VGA size. When an image is not scaled down (same size output), this restriction does not apply.

### 50.2.11 CFSZR, CFSZR\_x : Capture Filter Size Clip Register (x = B, M)

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0034 (CFSZR)  
 0x1034 (CFSZR\_B)  
 0x2034 (CFSZR\_M)



Bit	Symbol	Function	R/W
11:0	HFCLP[11:0]	Specify the horizontal clipping value of the filter output size (8-pixel units). The lower 3 bits should be written 0.	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W
27:16	VFCLP[11:0]	Specify the vertical clipping value of the filter output size (4-pixel units). The value of writing are ignored on the lower 2 bits, so write 0 to the lower 2 bits.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

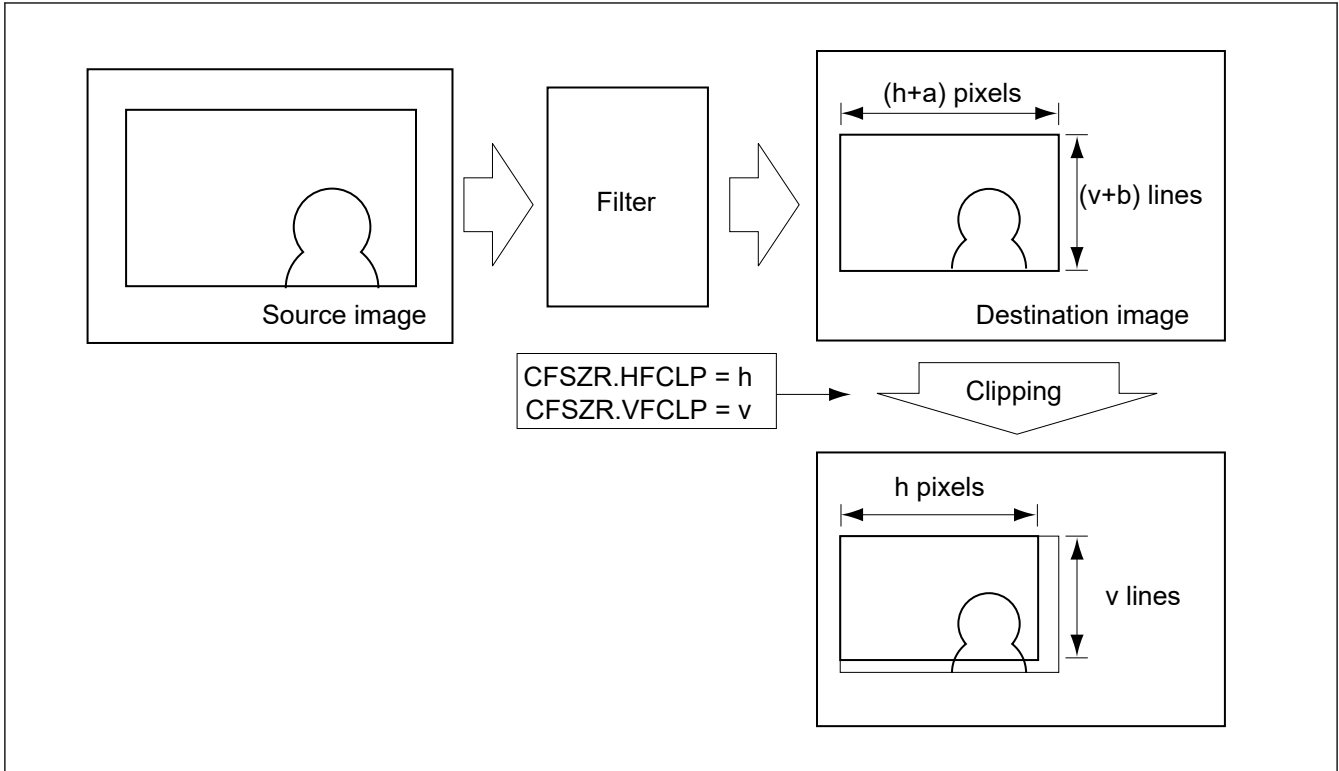
Note: S-TYPE-3, P-TYPE-3

CFSZR sets the clipping size for fine adjustment of the image size output from the filter, and must be set in combination with CFLCR. When clipping the output size of the filter, set the clipping size as a number of pixels, and the setting unit should be four pixels. CFSZR must be set even when scale-down is not performed (same size output).

This register is not used, during data enable fetch mode.

In data synchronous fetch mode, set CFSZR according to the setting of CAPWR.

The scale-down filter in the CEU may output an odd number of pixels or lines depending on the settings. To adjust the output size of the filter, the CEU clips the destination image by using the number of pixels specified in CFSZR, as shown in [Figure 50.33](#). The clipping size must be specified vertically in 4-pixel units and horizontally in 8-pixel units.



**Figure 50.33 Clipping of image output from filter**

The pixels to be clipped are counted from the top-left corner of a display. The pixels located to the right of the specified number of pixels or below the specified number of lines are discarded by the clipping function. If the number of pixels specified in CFSZR is larger than that output from the filter, correct operation cannot be guaranteed. To avoid this, the clipping size specified in CFSZR must be equal to or smaller than the number of pixels output from the filter.

**Note:** In data synchronous fetch mode, the following settings are required. Data cannot be fetched correctly unless the following settings are made.

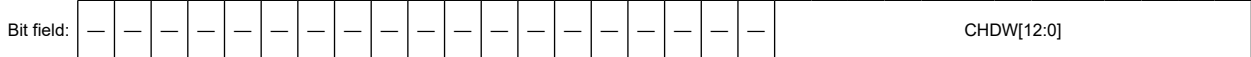
- VFCLP = CAPWR.VWDTH
- HFCLP = CAPWR.HWDTH/2

### 50.2.12 CDWDR, CDWDR\_x : Capture Destination Width Register (x = B, M)

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0038 (CDWDR)  
 0x1038 (CDWDR\_B)  
 0x2038 (CDWDR\_M)

Bit position: 31 12 0



Value after reset: 0

Bit	Symbol	Function	R/W
12:0	CHDW[12:0]	Specify the horizontal image size in the memory area where the captured image is to be stored (8-byte units). The image data captured by the CEU is stored in the memory. In data synchronous fetch mode, set as follows: CHDW = CAPWR.HWDTH The lower 3 bits should be written 0.	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

**Note:** S-TYPE-3, P-TYPE-3

CDWDR sets the horizontal image size in the memory area where the captured image is to be output in 8-byte units (8-pixel units).

In data synchronous fetch mode, set CDWDR according to the setting of CAPWR.

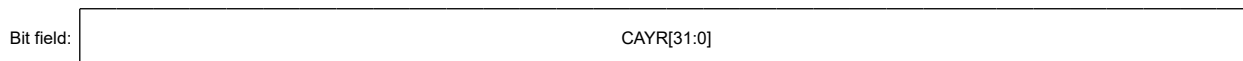
This register is not used, during data enable fetch mode.

### 50.2.13 CDAYR, CDAYR\_x : Capture Data Address Y Register (x = B, M)

Base address: CEU = 0x4034\_8000  
CEU\_NS = 0x5034\_8000

Offset address: 0x003C (CDAYR)  
0x103C (CDAYR\_B)  
0x203C (CDAYR\_M)

Bit position: 31 0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CAYR[31:0]	Capture Data Address Y <ul style="list-style-type: none"> <li>• Frame image capture: These bits set the address for storing the luminance component data of the captured data (8-pixel units).</li> <li>• One-field image capture: These bits set the address for storing the luminance component data of the captured data (8-pixel units).</li> <li>• Both-field image capture: These bits set the address for storing the luminance component data of the captured top-field data (8-pixel units).</li> <li>• Data fetch: These bits set the address for storing data (8-byte units).</li> <li>• Data enable fetch bundle write: These bits set the address for storing data (32-byte units).</li> </ul> The lower 3 bits should be written 0.	R/W

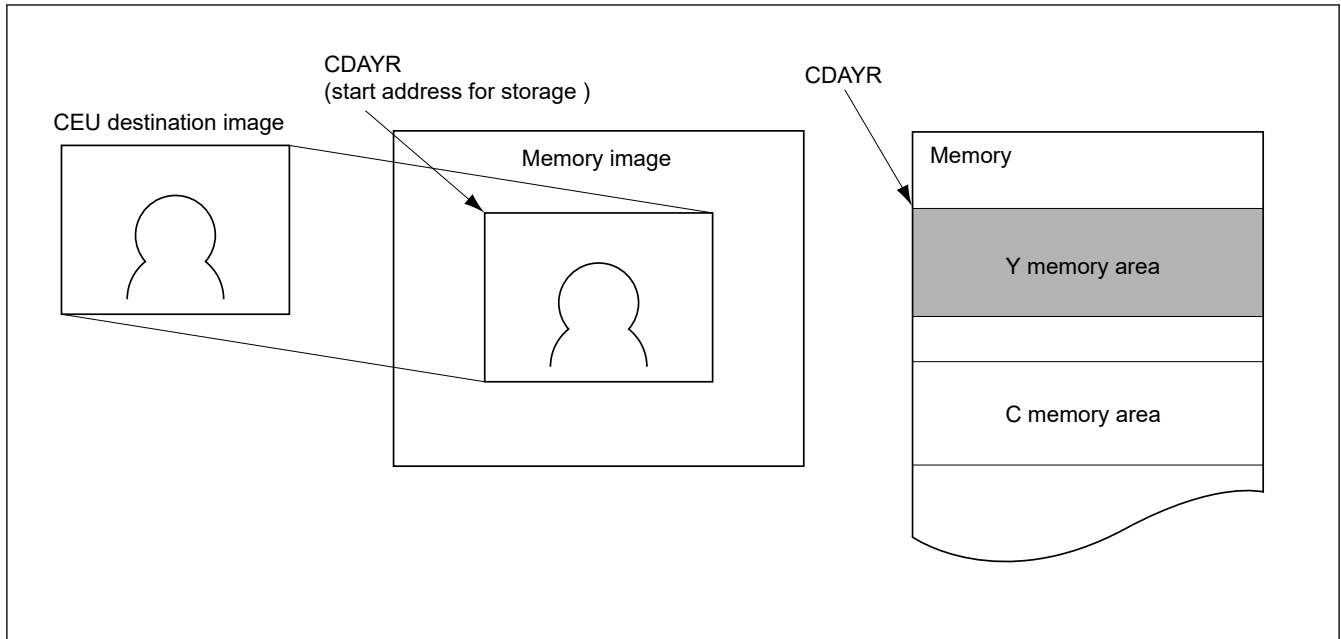
Note: S-TYPE-3, P-TYPE-3

CDAYR specifies the address where the luminance component of the captured data is to be stored in frame image capture or one-field image capture, the address where the luminance component of the captured top field is to be stored in both-field image capture, and the address where the fetched data is to be stored in data fetch. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the luminance component of the captured data is to be stored by CDAYR. In both-field image capture, set the start address of the memory area where the luminance component of the captured top-field image is to be stored by CDAYR. In data fetch, set the start address of the memory area where data is to be stored by CDAYR.

Because the address must be specified in 32 bits, the address set by CDAYR must be in longword units.

Set the address of the starting point of the memory area where the fetched data is to be stored in this register, as shown in [Figure 50.34](#).

- Frame image capture: Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the luminance component of the captured top-field image is to be stored.
- Data fetch: Set the address of the starting point of the memory area where the fetched data is to be stored. In data fetch mode, the data is simply stuffed in order from the start address so the end address becomes as follows:  
End address = CDAYR + number of fetched bytes
- Data enable fetch bundle write: Set the address in 32-byte units.



**Figure 50.34 Relationship between captured image and luminance component memory area**

### 50.2.14 CDACR, CDACR\_x : Capture Data Address C Register (x = B, M)

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0040 (CDACR)  
 0x1040 (CDACR\_B)  
 0x2040 (CDACR\_M)

Bit position: 31

0



Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CACR[31:0]	Capture Data Address C <ul style="list-style-type: none"> <li>Frame image capture: These bits set the address for storing the chrominance component data of the captured data (8-pixel units).</li> <li>One-field image capture: These bits set the address for storing the chrominance component data of the captured data (8-pixel units).</li> <li>Both-field image capture: These bits set the address for storing the chrominance component data of the captured top-field data (8-pixel units).</li> </ul> The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDACR specifies the address where the chrominance component of the captured data is to be stored in frame image capture or one-field image capture, and the address where the chrominance component of the captured top field is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or onefield image capture, set the start address of the memory area where the chrominance component of the captured data is to be stored by CDACR. In both-field image capture, set the start address of the memory area where the chrominance component of the captured top-field image is to be stored by CDACR. CDACR is not used in data fetch.

Because the address must be specified in 32 bits, the address set by CDACR must be in longword units.

Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored in this register, as shown in Figure 50.35. The chrominance component has an output data format like that in Figure 50.36, and is saved in the memory in this format.

- Frame image capture: Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored.

- One-field image capture: Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the chrominance component of the captured top-field image is to be stored.

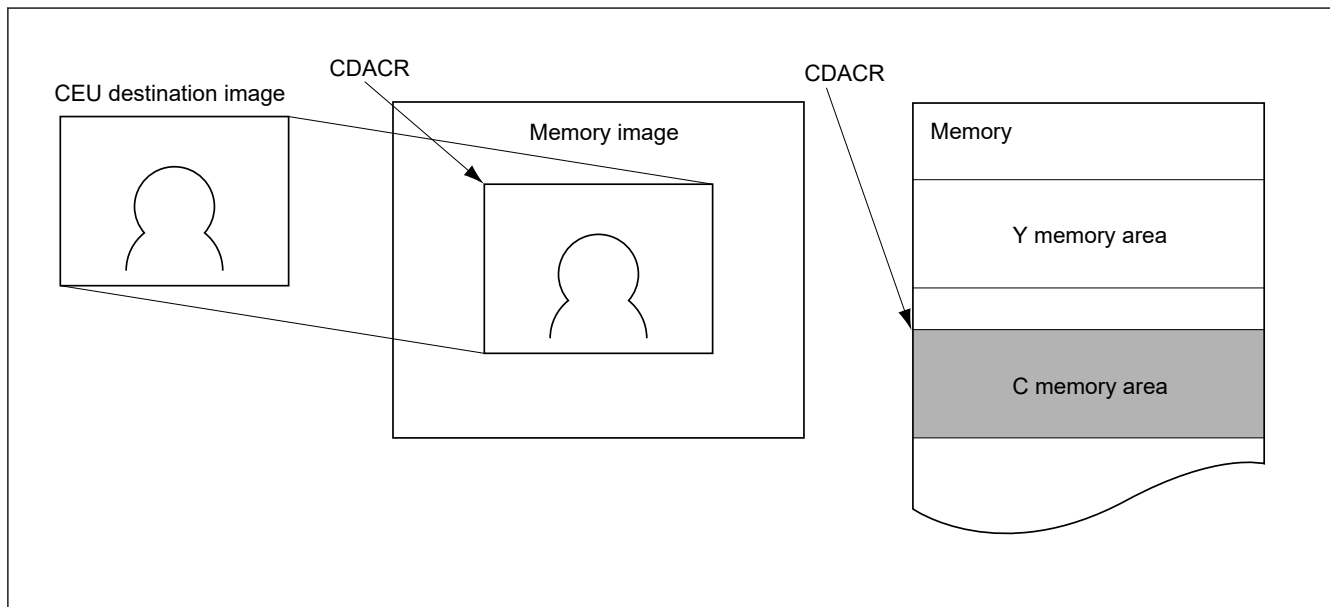


Figure 50.35 Relationship between captured image and chrominance component memory area

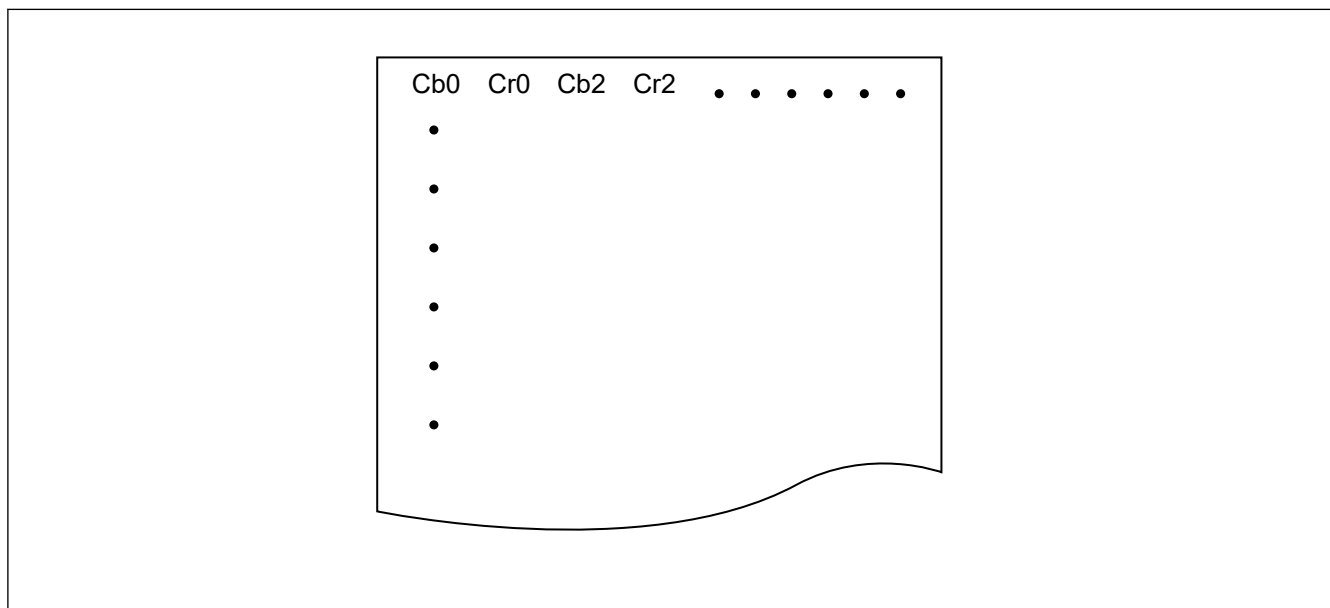


Figure 50.36 Image of storing chrominance components in memory

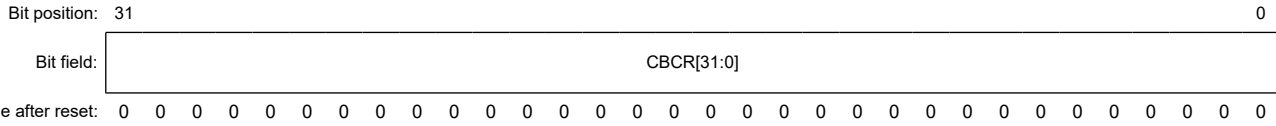




### 50.2.16 CDBCR, CDBCR\_x : Capture Data Bottom-Field Address C Register (x = B, M)

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0048 (CDBCR)  
 0x1048 (CDBCR\_B)  
 0x2048 (CDBCR\_M)



Bit	Symbol	Function	R/W
31:0	CBCR[31:0]	Set the address for storing the chrominance component data of the captured bottom-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDBCR specifies the address where the chrominance component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the chrominance component of the captured bottom-field image is to be stored by CDBCR. CDBCR is not used in frame image capture, one-field image capture, or data fetch.

Because the address must be specified in 32 bits, the address set by CDBCR must be in longword units.

Set the address of the starting point of the memory area where the chrominance component of the captured bottom-field image is to be stored in this register, as shown in Figure 50.38. The chrominance component has an output data format like that in Figure 50.39, and is saved in the memory in this format.

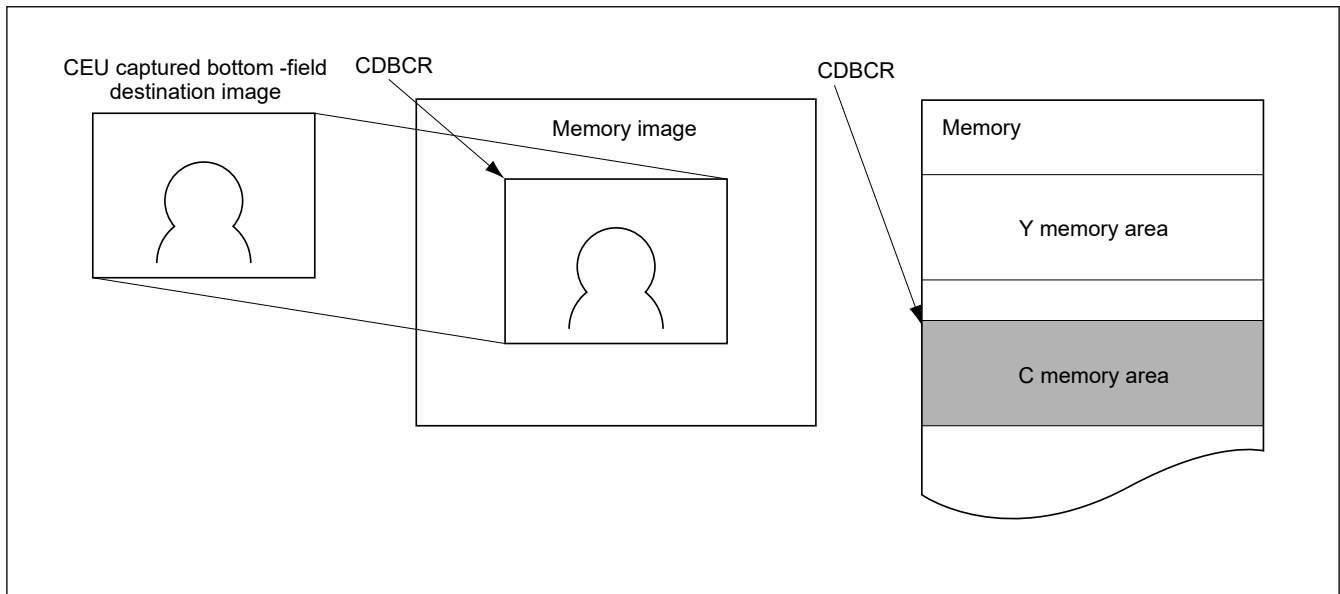
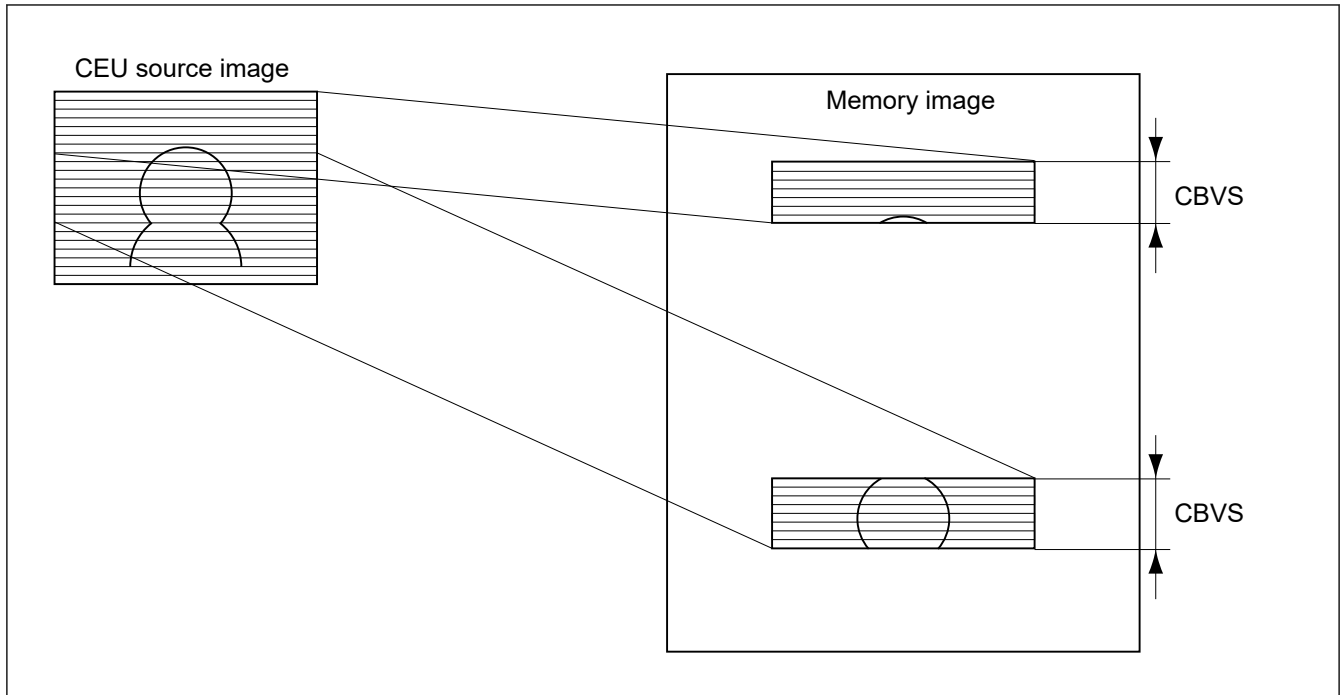


Figure 50.38 Relationship between captured bottom-field image and chrominance component memory area





**Figure 50.40 Image of storing captured image in memory by bundle write**

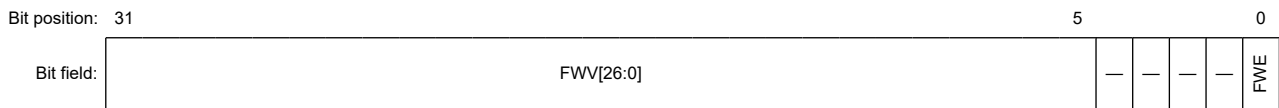
**(b) Data enable fetch**

Set the number of bytes of captured data to be written to the memory by a bundle write as a multiple of 32. This register is valid only when CDOCR.CBE = 1. The minimum settable size is 512 bytes. When a number smaller than 512 bytes is specified, operation is not guaranteed.

**50.2.18 CFWCR : Firewall Operation Control Register**

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x005C



Value after reset: 0 1 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
0	FWE	Firewall Operation With the setting of FWE = 1, when a write address exceeds the value set with FWV, the address is retained and an interrupt source FWF is set. After this, the address is not incremented and data is overwritten on the upper limit address. 0: Firewall is not activated. 1: Firewall is activated.	R/W
4:1	—	These bits are read as 0. The write value should be 0.	R/W
31:5	FWV[26:0]	Specify the upper limit of a write address. Specify the upper 27 bits of the 32-bit address. The upper limit of an address is FWV[26:0]<<5 + 0x1F.	R/W

Note: S-TYPE-3, P-TYPE-3

CFWCR specifies the upper limit of a write address in data enable fetch. When the VD input from an external module dose not go low and end notification is not given, this register can prevent writing to memory from being out of control.

This register is enabled only in data enable fetch.

### 50.2.19 CLFCR, CLFCR\_x : Capture Low-Pass Filter Control Register (x = B, M)

Base address: CEU = 0x4034\_8000  
CEU\_NS = 0x5034\_8000

Offset address: 0x0060 (CLFCR)  
0x1060 (CLFCR\_B)  
0x2060 (CLFCR\_M)

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

Bit	Symbol	Function	R/W
0	LPF	Enables or disables operation of the low-pass filter. The low-pass filter removes high-frequency components from the destination image in the horizontal direction. Clear this bit to 0 in data fetch mode. 0: Low-pass filter not used 1: Low-pass filter used (only in the horizontal direction)	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CLFCR specifies whether or not to operate the low-pass filter. In data fetch mode, clear the LPF bit to 0.

The characteristic of the low-pass filter installed in the CEU causes the phase location of the image processed by the low-pass filter to be shifted right by one pixel compared to the raw image.

### 50.2.20 CDOCR, CDOCR\_x : Capture Data Output Control Register (x = B, M)

Base address: CEU = 0x4034\_8000  
CEU\_NS = 0x5034\_8000

Offset address: 0x0064 (CDOCR)  
0x1064 (CDOCR\_B)  
0x2064 (CDOCR\_M)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CDS	—	COLS	COWS	COBS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	COBS	Controls swapping in 8-bit units for data output from the CEU. 0: Data is not swapped in 8-bit units 1: Data is swapped in 8-bit units	R/W
1	COWS	Controls swapping in 16-bit units for data output from the CEU. 0: Data is not swapped in 16-bit units 1: Data is swapped in 16-bit units	R/W
2	COLS	Controls swapping in 32-bit units for data output from the CEU. 0: Data is not swapped in 32-bit units 1: Data is swapped in 32-bit units	R/W

Bit	Symbol	Function	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	CDS	<p>Sets the image format when outputting the image data captured in the YCbCr422 format to the memory.</p> <p>When 0 is written to this bit, only the luminance component (Y) is output and no chrominance components (Cb and Cr) are output for the odd-numbered lines. With an interlace source image, similarly only the luminance component (Y) is output and no chrominance components (Cb and Cr) are output for the odd-numbered lines of the field. In data fetch mode, set this bit to 1.</p> <p>0: Converts the YCbCr422 format to the YCbCr420 format before outputting data to the memory 1: Outputs data in the YCbCr422 format to the memory without conversion</p>	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
16	CBE	<p>Controls the number of lines of captured data to be written to the memory.</p> <ul style="list-style-type: none"> <li>• Image capture This bit controls the number of lines of captured data to be written to the memory. When bundle write is set by this register, captured data is written in line units specified by CBDSR to the addresses specified by CDAYR and CDACR, and CDAYR2 and CDACR2 (CDBYR and CDBCR, and CDBYR2 and CDBCR2 for the bottom field in both-field capture) alternately (Figure 50.41). When captured data has been written for the number of lines set by CBDSR, a write end interrupt corresponding to each address setting register occurs. However, after write for one-frame (one-field) capture ends, a bundle write end interrupt does not occur even when bundle write has finished.</li> <li>• Data synchronous fetch This bit controls the number of lines of captured data to be written to the memory. When bundle write is set by this register, captured data is written in line units specified by CBDSR to the addresses specified by CDAYR and CDAYR2 alternately. When captured data has been written for the number of lines set by CBDSR, a write end interrupt corresponding to each address setting register occurs. However, after write for one-frame capture ends, a bundle write end interrupt does not occur even when bundle write has finished.</li> <li>• Data enable fetch This bit controls the number of bytes of captured data to be written to the memory. When bundle write is set by this register, captured data is written in byte units specified by CBDSR to the addresses specified by CDAYR and CDAYR2 alternately. When captured data has been written for the number of bytes set by CBDSR, a write end interrupt corresponding to each address setting register occurs. Also, only in data enable fetch, a bundle write end interrupt occurs when bundle write has finished after write for one-frame capture ends.</li> </ul> <p>Table 50.9 shows the correspondence between address setting registers and write end interrupt sources. Figure 50.42 shows the timing of write end interrupts in image capture and data synchronous fetch. Figure 50.43 shows the timing of write end interrupts in data enable fetch.</p> <p>0: Normal write 1: Bundle write</p>	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDOCR sets the format for outputting captured data to the memory. In data fetch mode, set the CDS bit to 1.

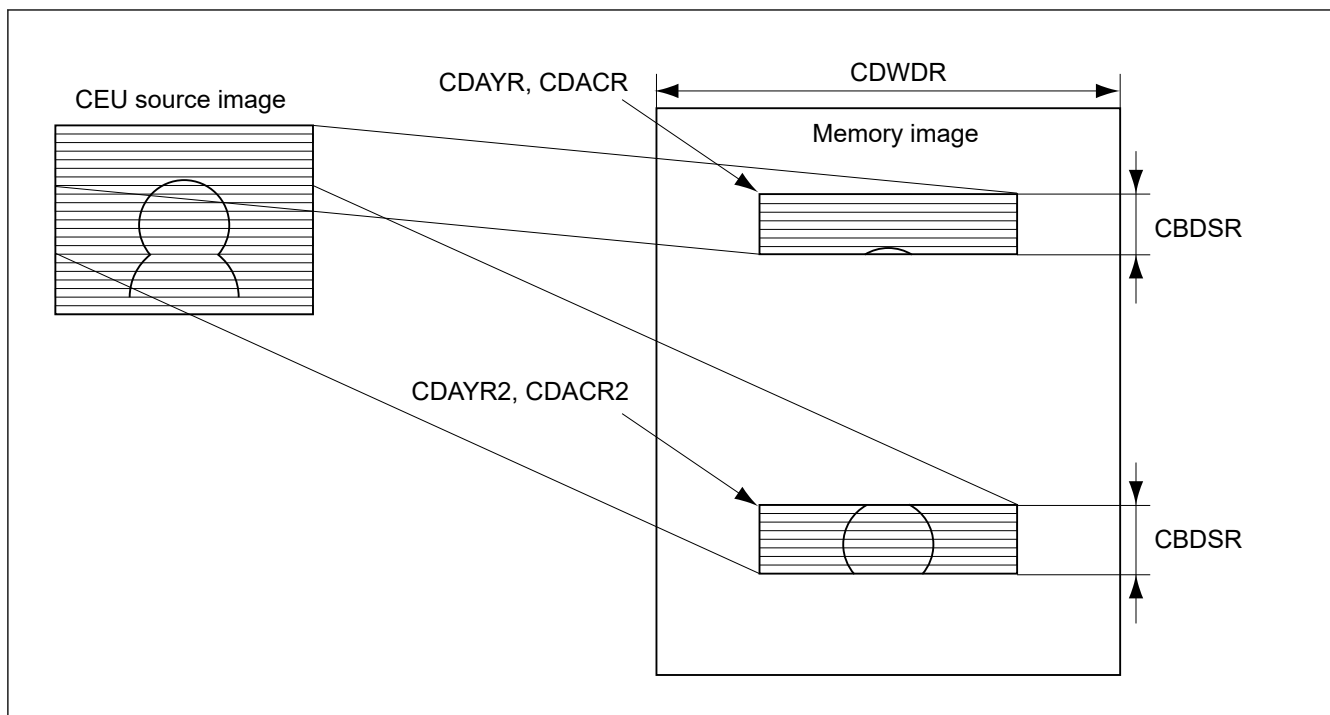


Figure 50.41 Image of bundle write to memory

Table 50.9 Correspondence between address setting registers and write end interrupt sources

Address setting registers	Bundle write end interrupt source
CDAYR, CDACR	CPBE1 bit in CETCR
CDAYR2, CDACR2	CPBE2 bit in CETCR
CDBYR, CDBCR	CPBE3 bit in CETCR
CDBYR2, CDBCR2	CPBE4 bit in CETCR

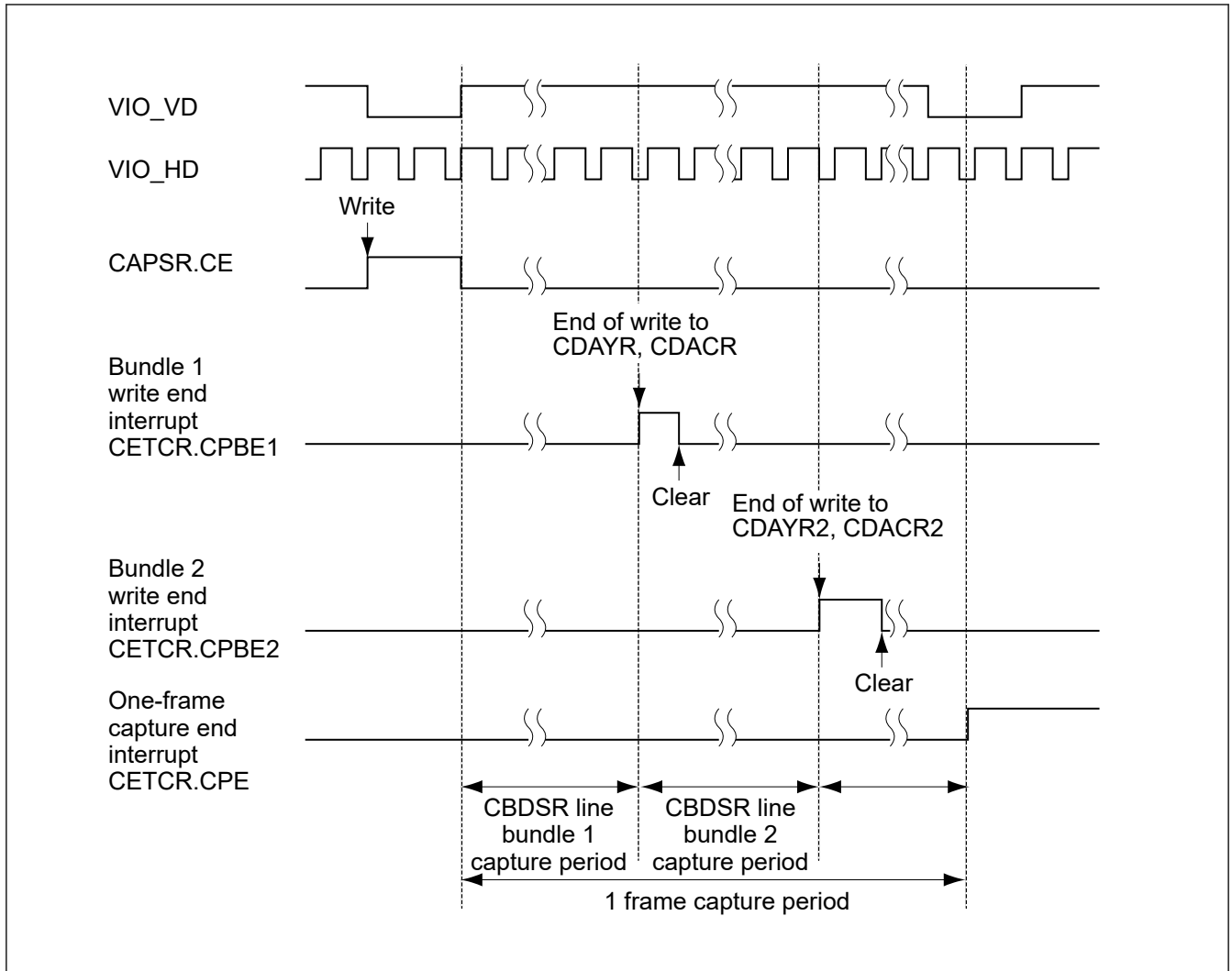
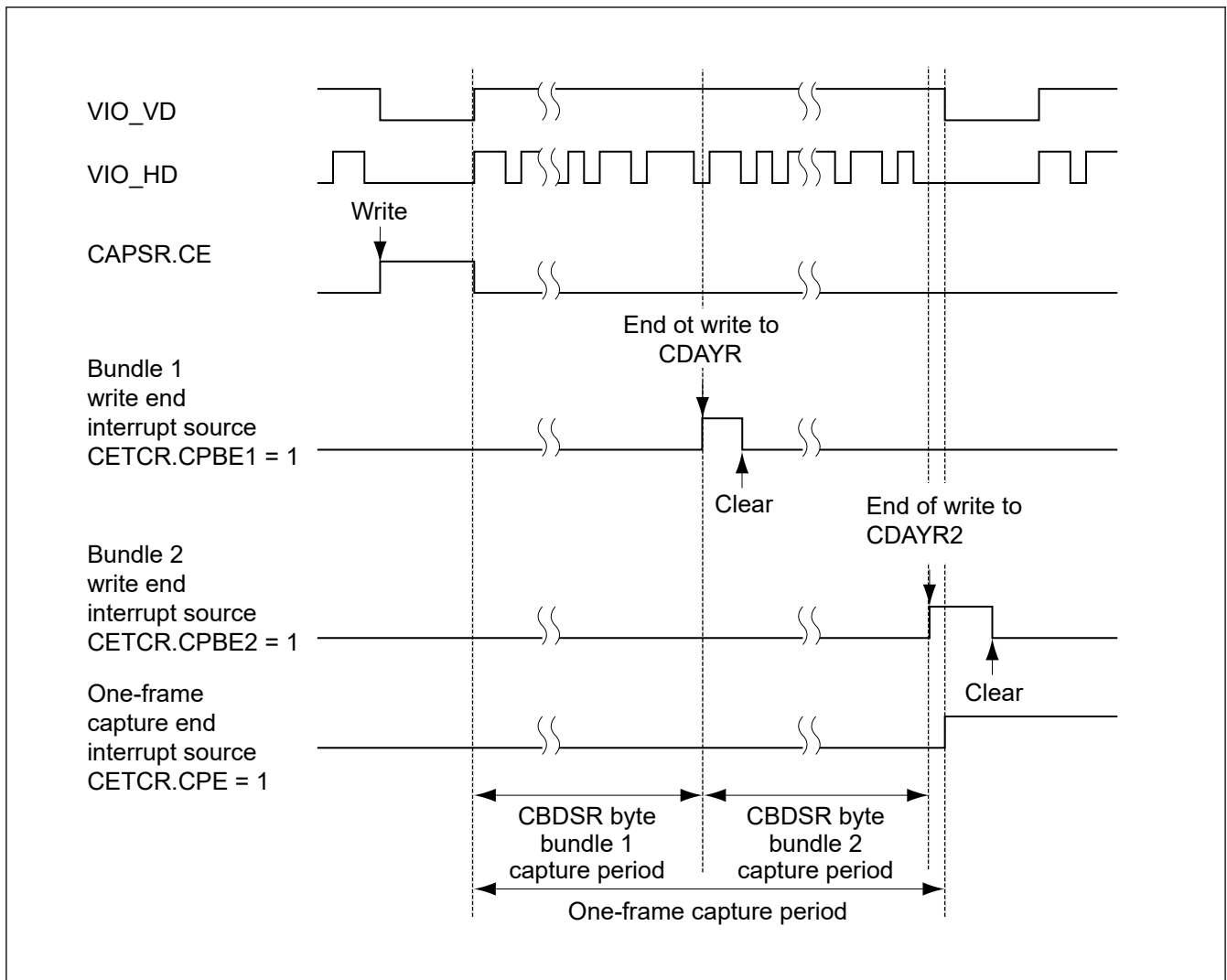


Figure 50.42 Timing of write end interrupts (Image capture, data synchronous fetch)



**Figure 50.43 Timing of write end interrupts (data enable fetch)**

For data output from the CEU, the COLS, COWS, and COBS bits control swapping in 32-bit, 16-bit, and 8-bit units, respectively. Set these bits when data is misaligned because of a difference in endian. The data swapping bits are shown below. These bits can be set similarly in data fetch mode.

Data can be swapped in 8-bit, 16-bit, 32-bit units, or in 32 bits, 16 bits and 8 bits, as shown in [Figure 50.44](#). To enable data swapping, set the corresponding control bit to 1.



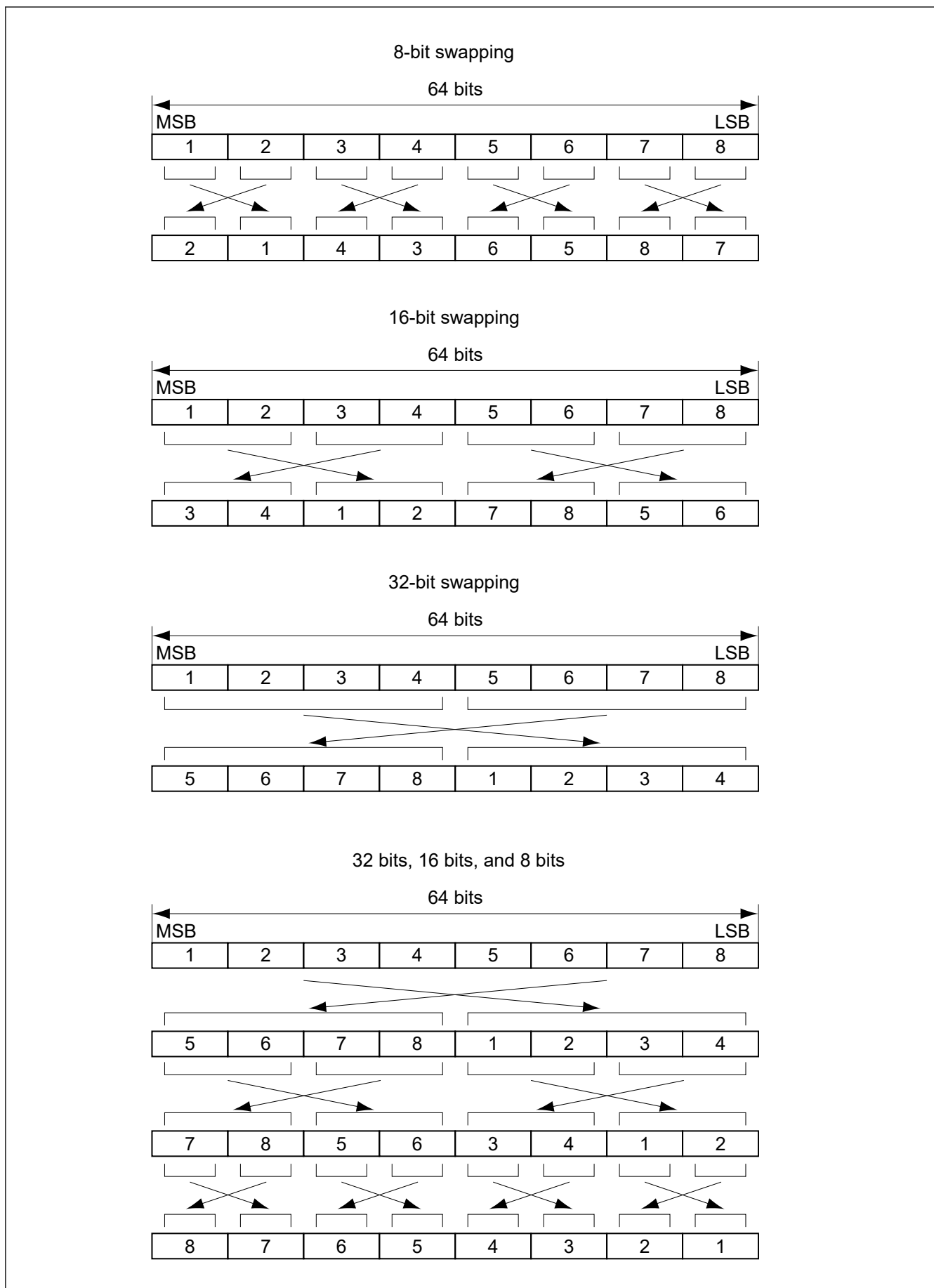


Figure 50.44 Data swapping by data aligner

### 50.2.21 CEIER : Capture Event Interrupt Enable Register

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	NVDIE	NHDIE	FWFIE	—	—	VBPIE	—	IGVSI E	IGHSI E	CDTO FIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CPBE 4IE	CPBE 3IE	CPBE 2IE	CPBE 1IE	—	—	VDIE	HDIE	—	—	—	IGRWI E	—	—	CFEIE	CPEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPEIE	One-Frame Capture End Interrupt Enable 0: Disables a one-frame capture end interrupt 1: Enables a one-frame capture end interrupt	R/W
1	CFEIE	CFE Interrupt Enable 0: Disables a CFE interrupt 1: Enables a CFE interrupt	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	IGRWIE	Register-Access-During-Capture Interrupt Enable 0: Disables a register-access-during-capture interrupt 1: Enables a register-access-during-capture interrupt	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	HDIE	HD Interrupt Enable 0: Disables an HD interrupt 1: Enables an HD interrupt	R/W
9	VDIE	VD Interrupt Enable 0: Disables a VD interrupt 1: Enables a VD interrupt	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	CPBE1IE	CPBE1 Interrupt Enable 0: Disables a CPBE1 interrupt 1: Enables a CPBE1 interrupt	R/W
13	CPBE2IE	CPBE2 Interrupt Enable 0: Disables a CPBE2 interrupt 1: Enables a CPBE2 interrupt	R/W
14	CPBE3IE	CPBE3 Interrupt Enable 0: Disables a CPBE3 interrupt 1: Enables a CPBE3 interrupt	R/W
15	CPBE4IE	CPBE4 Interrupt Enable 0: Disables a CPBE4 interrupt 1: Enables a CPBE4 interrupt	R/W
16	CDTOFIE	CDTOF Interrupt Enable 0: Disables a CDTOF interrupt 1: Enables a CDTOF interrupt	R/W
17	IGHSE	IGHS Interrupt Enable 0: Disables an IGHS interrupt 1: Enables an IGHS interrupt	R/W
18	IGVSE	IGVS Interrupt Enable 0: Disables an IGVS interrupt 1: Enables an IGVS interrupt	R/W

Bit	Symbol	Function	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	VBPIE	VBP Interrupt Enable 0: Disables a VBP interrupt 1: Enables a VBP interrupt	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	FWFIE	FWF Interrupt Enable 0: Disables a FWF interrupt 1: Enables a FWF interrupt	R/W
24	NHDIE	Non-HD Interrupt Enable Disable this interrupt (NHDIE = 0) for data enable fetch. 0: Disables a non-HD interrupt 1: Enables a non-HD interrupt	R/W
25	NVDIE	Non-VD Interrupt Enable Disable this interrupt (NVDIE = 0) for data enable fetch. 0: Disables a non-VD interrupt 1: Enables a non-VD interrupt	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CEIER enables or disables interrupts of the event flag register that generates CEU interrupts.

### 50.2.22 CETCR : Capture Event Flag Clear Register

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0074

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	NVD	NHD	FWF	—	—	VBP	—	IGVS	IGHS	CDTO F
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CPBE 4	CPBE 3	CPBE 2	CPBE 1	—	—	VD	HD	—	—	—	IGRW	—	—	CFE	CPE
Value after reset:	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPE	An interrupt indicating that capturing of one frame from an external module has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next VD input. This interrupt indicates that capturing of one frame has finished. This bit is set to 1 when the image of the size set in CAPWR is captured and the last data transfer to the bus finished (see <a href="#">Figure 50.46</a> ).	R/W
1	CFE	An interrupt indicating that capturing of one field from an external module has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next VD input (see <a href="#">Figure 50.45</a> ). This interrupt occurs only in both-field capture mode.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	IGRW	An interrupt indicating that during capturing, access was attempted to a register to which writing during operation is prohibited. Among the CEU registers, writing during capturing is prohibited for some registers. <a href="#">Table 50.10</a> shows which registers can/cannot be written to during capturing. This bit is set to 1 when a register to which writing during capturing is prohibited has been written to.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	HD	An interrupt indicating that HD (horizontal sync signal) was input from an external module. This bit is set to 1 when an HD input from an external module is detected. Immediately after the HDPOL bit in CAMCR is modified, a pseudo HD is input and this bit is set to 1. The HD interrupt after the HDPOL bit is modified should be ignored.	R/W
9	VD	An interrupt indicating that VD (vertical sync signal) was input from an external module. In data enable fetch mode, this bit is set to 1 when a VD input from an external module is detected. In image capture mode and data synchronous fetch mode, this module generates a VD interrupt in response to the first detection of the active level of HD following detection of the active level of VD from an external module. Note that, when VD and HD are asserted and detected at the same time, this module generates a VD interrupt at that time. Immediately after the VDPOL bit in CAMCR is modified, a pseudo VD is input and this bit is set to 1. The VD interrupt after the VDPOL bit is modified should be ignored.	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	CPBE1	An interrupt indicating that writing to CDAYR and CDACR in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines (number of bytes in data enable fetch) set in CBDSR has been captured and the last data transfer to the bus has completed. However, in image capture or data synchronous fetch, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).	R/W
13	CPBE2	An interrupt indicating that writing to CDAYR2 and CDACR2 in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines (number of bytes in data enable fetch) set in CBDSR has been captured and the last data transfer to the bus has completed. However, in image capture or data synchronous fetch, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).	R/W
14	CPBE3	An interrupt indicating that writing to CDBYR and CDBCR in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines set in CBDSR has been captured and the last data transfer to the bus has completed. However, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).	R/W
15	CPBE4	An interrupt indicating that writing to CDBYR2 and CDBCR2 in a bundle write has finished. This interrupt is output when the last captured data has been transferred and the end notification received, regardless of the next HD input. This bit is set to 1 when data for the number of lines set in CBDSR has been captured and the last data transfer to the bus has completed. However, this interrupt does not occur when the last captured data in a bundle write is the last captured data of a frame (field).	R/W
16	CDTOF	An interrupt indicating that data overflowed in the CRAM of the write buffer. Since data is input at realtime from an external module in capture operations, the frame image is overwritten unless the captured data is transferred from the CEU internal buffer to the memory at a certain or higher transfer rate. This bit is set to 1 when writing the data in the CRAM of the CEU internal write buffer to the bus is not performed within time and data has overflowed.	R/W
17	IGHS	An interrupt generated when the number of HD cycles set in CMCYR differ from the number of HD cycles input from an external module. This bit is set to 1 when there is an illegal HD input from an external module. This bit is set to 1 when the number of clock cycles for the HD input to the CEU differs from the value set in the HCYL bits in CMCYR. Note however that when the HCYL bits are cleared to 0, this interrupt does not occur.	R/W
18	IGVS	An interrupt generated when the number of VD cycles set in CMCYR differ from the number of VD cycles input from an external module. This bit is set to 1 when there is an illegal VD input from an external module. This bit is set to 1 when the number of HD cycles for the VD input to the CEU differs from the value set in the VCYL bits in CMCYR. Note however that when the VCYL bits are cleared to 0, this interrupt does not occur.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20	VBP	An interrupt indicating that VD has been input while the CEU holds data (insufficient vertical-sync front porch). The conditions for a VBP interrupt to occur are as follows: <ul style="list-style-type: none"> <li>Condition 1 VD is input when there is captured data within the CEU</li> <li>Condition 2 The last transfer data cannot be internally detected due to a write buffer overflow or an illegal HD so that the end timing is unclear until the next VD (By generating a VBP interrupt at the VD input timing, capture fail can be announced.)</li> </ul> When a VBP interrupt occurs, a capture end interrupt (CPE bit in CETCR) does not occur and the image of that frame is not captured correctly. Though a capture end interrupt (CPE bit) will occur on rare occasions, it should be ignored in this case. Capturing cannot be performed until the next VD (even if the CE bit (capture reservation signal) in CAPSR is 1, capturing does not start). In the case of condition 2, instead of waiting for a VBP interrupt to occur, execute a software reset (CPKIL bit in CAPSR) to stop capturing and then restart capturing. In this case, since capture operation is terminated without waiting for the next VD, a VBP interrupt does not occur and capturing can be performed from the next VD.	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	FWF	The interrupt is generated when data is written to the address that exceeds the value specified with CFWCR.FMV. This bit is set to 1 when data is written to the address that exceeds the value specified with CFWCR.FMV while CFWCR.FWE = 1.	R/W
24	NHD	An interrupt indicating that no HD was input. The timing for a non-HD interrupt to occur differs depending on the bit width of the digital image input pins. Occurs when the 11-bit internal counter that is incremented every eight cycles becomes full. Accordingly, this bit is set to 1 when no HD has been input for at least 16,376 cycles since the last HD was input. When connecting a camera whose HD is fixed low when VD is low, this bit may be set to 1. Ignore this interrupt during data enable fetch.	R/W
25	NVD	An interrupt indicating that no VD was input. A non-VD interrupt occurs when the 14-bit internal counter becomes full. Accordingly, this bit is set to 1 when no VD has been input for at least 16,383 lines since the last VD was input.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CETCR notifies the CPU of the source of an interrupt that is generated in the CEU. The flags of CETCR can be used as interrupt signals. When the corresponding interrupt is enabled, an interrupt (CEU\_CEUI) is generated. To clear the interrupt, clear the bit corresponding to the interrupt source to 0. After several cycles have passed after modifying the bit, the interrupt is cleared.

To clear the bit corresponding to the interrupt source to be cleared to 0 and retain that state, write 1 to that bit. For example, to clear only the CPE bit to 0, write 0xFFFFFFE to CETCR.

In CETCR, only bits to which 0 is written are cleared. Bits to which 1 is written retain their current values. To clear an interrupt source, write 0 only to the bit corresponding to the interrupt source to be cleared, and 1 to the other bits.

Note: Since the CETCR value becomes undefined in the following cases, clear all bits in CETCR to 0.

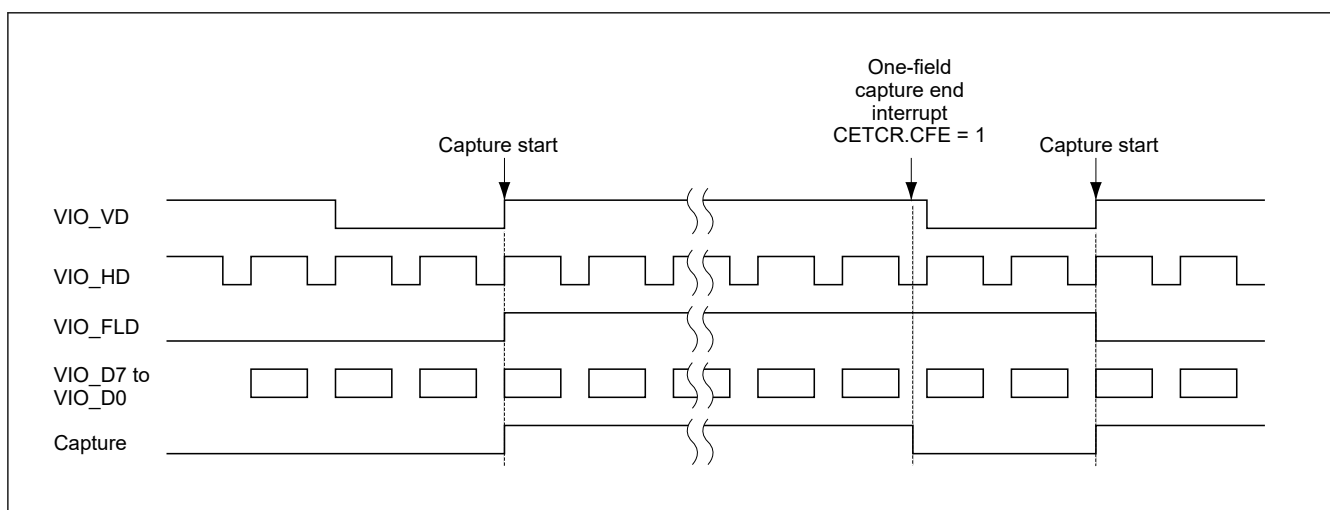
- VD and HD bits immediately after system reset or Software Standby mode is entered.
- VD and HD bits after the polarities of the capture interface sync signals are changed.

**Table 50.10 Registers that can/cannot be modified during capturing (1 of 2)**

Register name	Modification during capturing
CAPSR	Possible
CAPCR	Prohibited
CAMCR	Prohibited
CMCYR	Prohibited
CAMOR	Possible

**Table 50.10 Registers that can/cannot be modified during capturing (2 of 2)**

Register name	Modification during capturing
CAPWR	Possible
CAIFR	Prohibited
CRCNTR	Possible
CRCMPR	Prohibited
CFLCR	Possible
CFSZR	Possible
CDWDR	Possible
CDAYR	Possible
CDACR	Possible
CDBYR	Possible
CDBCR	Possible
CBDSR	Possible
CFWCR	Possible
CLFCR	Possible
CDOCR	Possible
CEIER	Possible
CETCR	Possible
CSTSR	Prohibited
CDSSR	Prohibited
CDAYR2	Possible
CDACR2	Possible
CDBYR2	Possible
CDBCR2	Possible



**Figure 50.45 CFE generation timing**

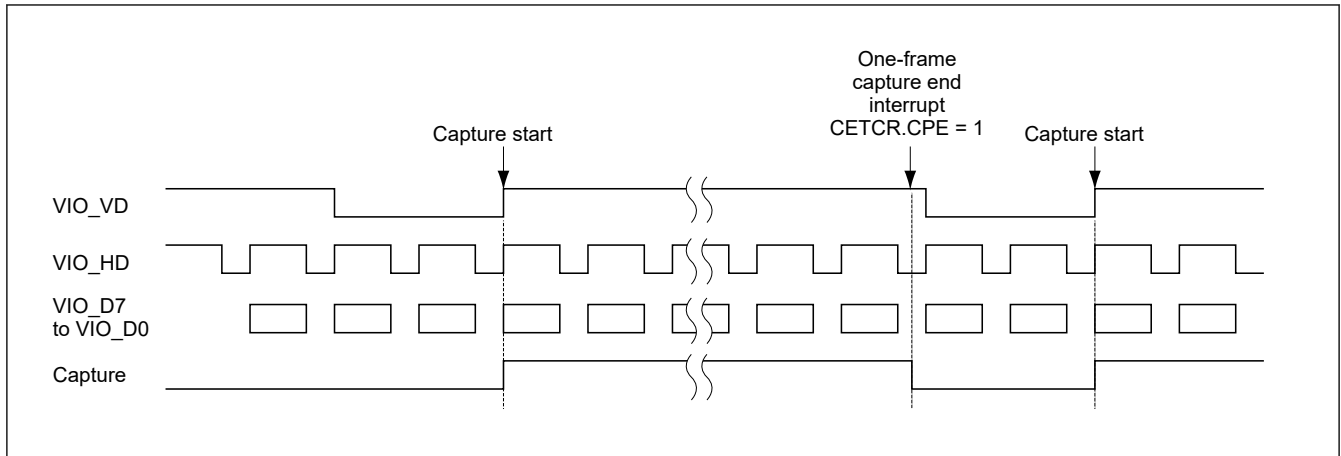


Figure 50.46 CPE generation timing

### 50.2.23 CSTSR : Capture Status Register

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x007C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	CRST	—	—	—	—	—	—	—	CPFLD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CPTON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPTON	Indicates that the CEU is operating. This bit retains 1 during the period that starts from the internal VD at capture start and ends when a one-frame capture end interrupt occurs. Figure 50.47 shows the CEU operating period.	R
15:1	—	These bits are read as 0.	R
16	CPFLD	Indicates which field is being captured. 0: Bottom field is being captured 1: Top field is being captured	R
23:17	—	These bits are read as 0.	R
24	CRST	Indicates which register plane is currently used. 0: Plane A of the register is being used 1: Plane B of the register is being used	R
31:25	—	These bits are read as 0.	R

Note: S-TYPE-3, P-TYPE-3

CSTSR indicates the internal status of the CEU. CSTSR differs from CETCR in that no interrupt is generated for the events indicated in CSTSR.

The CEU operating/halt state can be confirmed using CSTSR. To confirm the halt state of the CEU, make sure that the status bit (bit 0) indicating that the CEU is operating is cleared to 0 for sure.

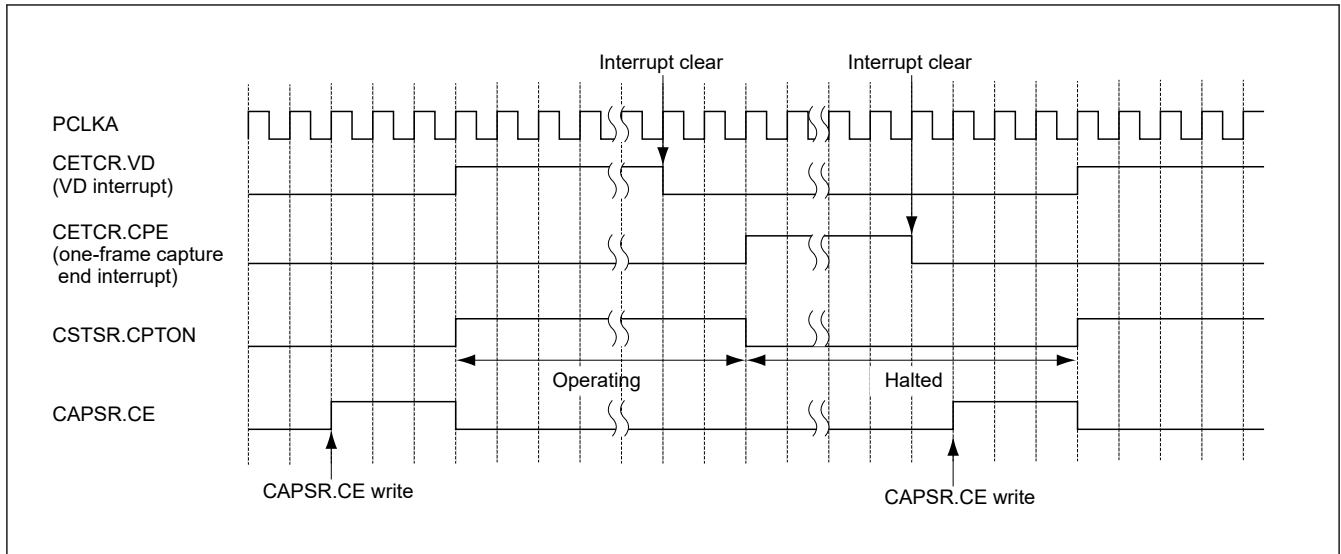


Figure 50.47 Operating status during capturing

### 50.2.24 CDSSR : Capture Data Size Register

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0084

Bit position: 31

0

Bit field:

CDSS[31:0]

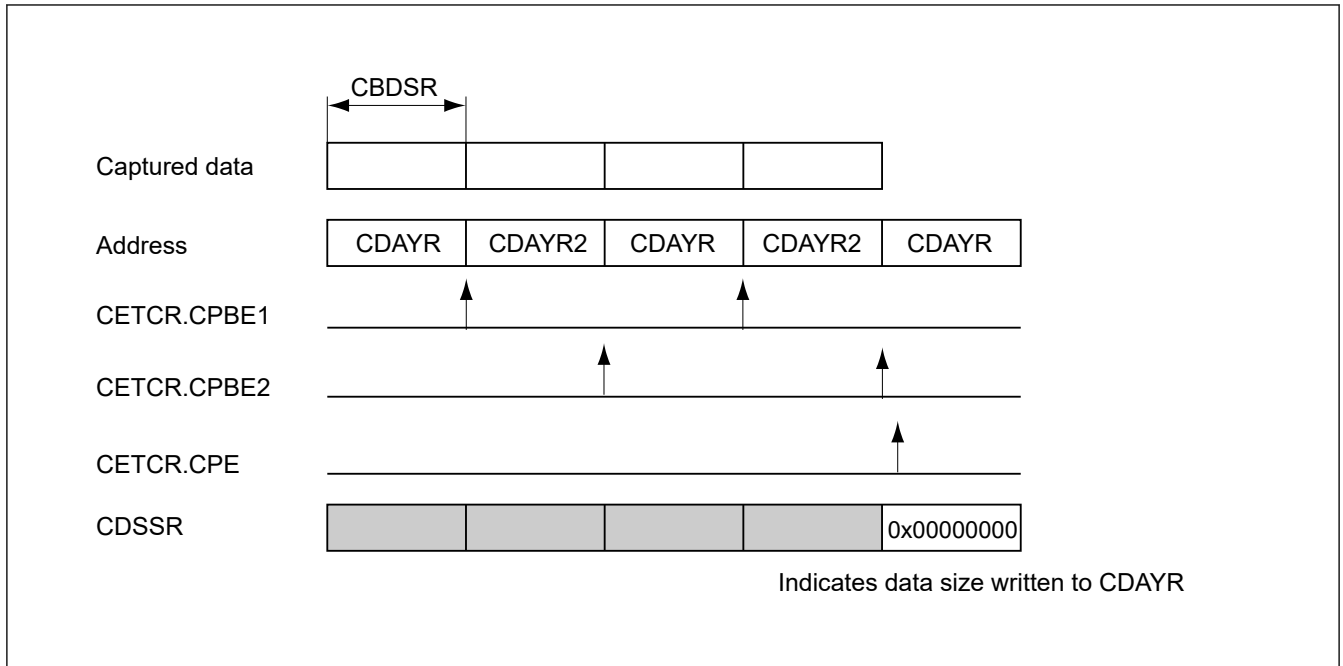
Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CDSS[31:0]	Indicate the size of data written to the memory in data enable fetch. In a bundle write, size of data written to the selected address at the end of one-frame capture is indicated. In a bundle write, as soon as the number of bytes specified by CBDSR is transferred to the bus, address to which data is written is switched. Therefore, if one-frame capture is completed at the same time as a bundle write is completed, this register indicates 0x00000000. <a href="#">Figure 50.48</a> and <a href="#">Figure 50.49</a> show the overall timing of the CDSSR operation in a bundle write.	R

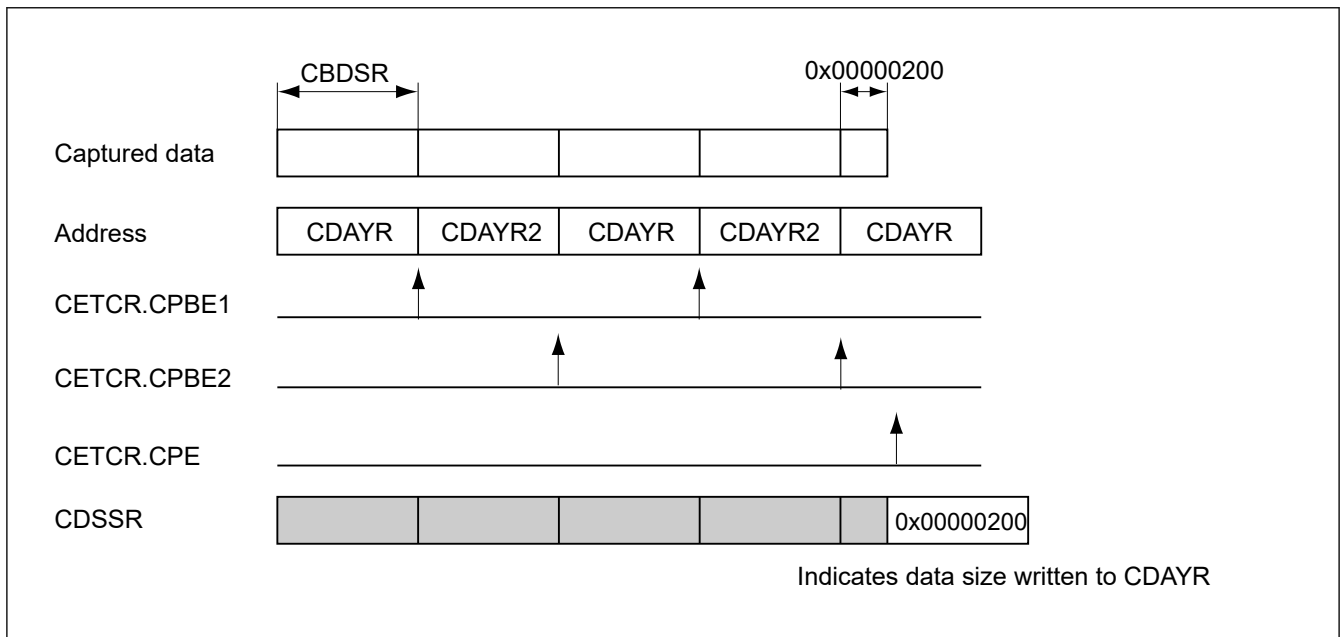
Note: S-TYPE-3, P-TYPE-3

CDSSR indicates the size of data written to the memory in data enable fetch. As this register indicates a correct value at the end of capture, confirm this register when capture is completed.





**Figure 50.48 Overall timing of CDSSR operation in bundle write (when bundle write end and capture end coincide)**

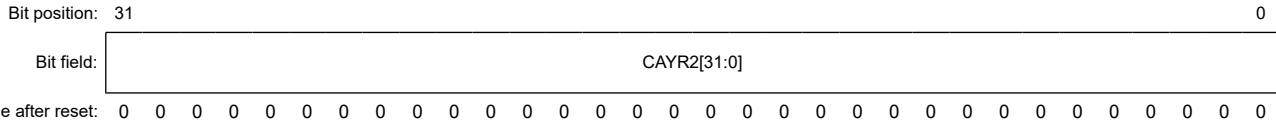


**Figure 50.49 Overall timing of CDSSR operation in bundle write (when bundle write end and capture end do not coincide)**

### 50.2.25 CDAYR2, CDAYR2\_x : Capture Data Address Y Register 2 (x = B, M)

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0090 (CDAYR2)  
 0x1090 (CDAYR2\_B)  
 0x2090 (CDAYR2\_M)



Bit	Symbol	Function	R/W
31:0	CAYR2[31:0]	Capture Data Address Y Frame image capture: These bits set the address for storing the luminance component data of the captured data (8-pixel units). <ul style="list-style-type: none"> <li>• One-field image capture: These bits set the address for storing the luminance component data of the captured data (8-pixel units).</li> <li>• Both-field image capture: These bits set the address for storing the luminance component data of the captured top-field data (8-pixel units).</li> <li>• Data synchronous fetch: These bits set the address for storing data (8-byte units).</li> <li>• Data enable fetch: These bits set the address for storing data (32-byte units).</li> </ul> The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

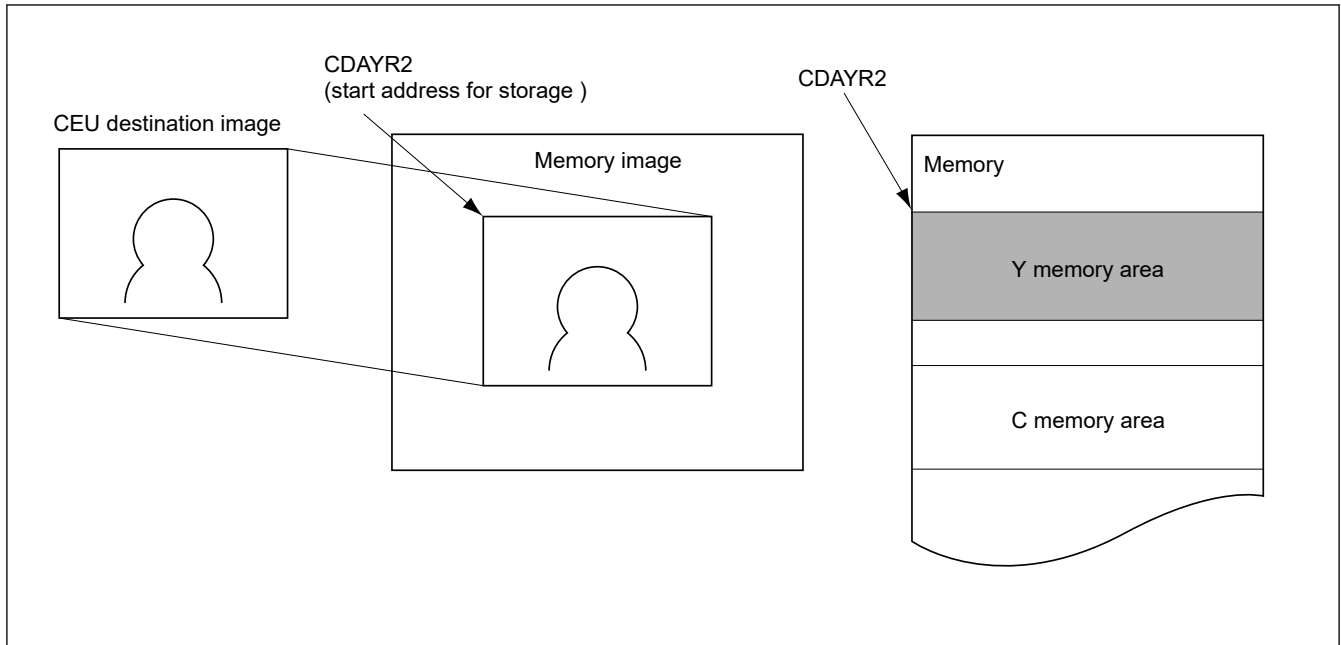
CDAYR2 specifies the address for the luminance component used in a bundle write and the address for data storage in a bundle write in data fetch. CDAYR2 is used only in a bundle write.

CDAYR2 specifies the address where the luminance component of the captured data is to be stored in frame image capture or one-field image capture, the address where the luminance component of the captured top field is to be stored in both-field image capture, and the address where data is to be stored in data fetch. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the luminance component of the captured data is to be stored by CDAYR2. In both-field image capture, set the start address of the memory area where the luminance component of the captured top-field image is to be stored by CDAYR2. In data fetch, set the start address of the memory area to be used for data storage.

The address specified by this register must be in 32-bit units.

Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored in this register, as shown in [Figure 50.50](#).

- Frame image capture: Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the luminance component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the luminance component of the captured top-field image is to be stored.
- Data synchronous fetch: Set the address of the starting point of the memory area where the captured data is to be stored.
- Data enable fetch: Set the address of the starting point of the memory area where the captured data is to be stored in 32-byte units.



**Figure 50.50 Relationship between captured image and luminance component memory area**

**50.2.26 CDACR2, CDACR2\_x : Capture Data Address C Register 2 (x = B, M)**

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0094 (CDACR2)  
 0x1094 (CDACR2\_B)  
 0x2094 (CDACR2\_M)

Bit position: 31 0

Bit field: CACR2[31:0]

Value after reset: 0

Bit	Symbol	Function	R/W
31:0	CACR2[31:0]	Capture Data Address C <ul style="list-style-type: none"> <li>Frame image capture: These bits set the address for storing the chrominance component data of the captured data (8-pixel units).</li> <li>One-field image capture: These bits set the address for storing the chrominance component data of the captured data (8-pixel units).</li> <li>Both-field image capture: These bits set the address for storing the chrominance component data of the captured top-field data (8-pixel units).</li> </ul> The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDACR2 specifies the address for the chrominance component used in a bundle write. CDACR2 is used only in a bundle write.

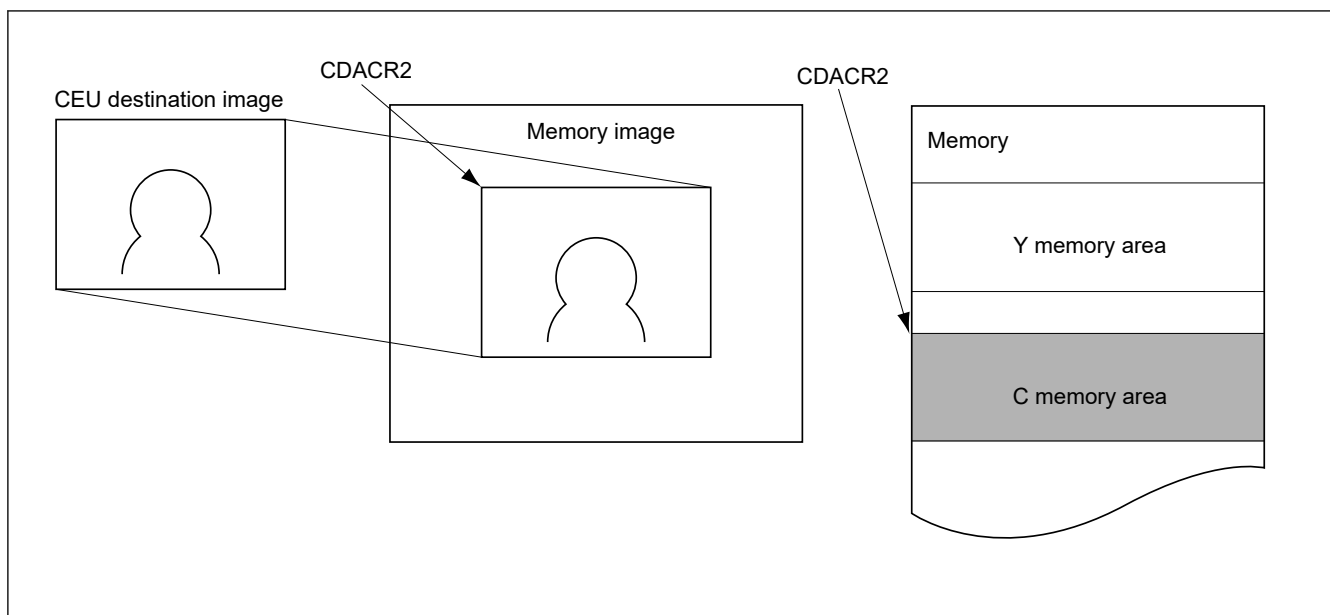
CDACR2 specifies the address where the chrominance component of the captured data is to be stored in frame image capture or one-field image capture, and the address where the chrominance component of the captured top field is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. In frame image capture or one-field image capture, set the start address of the memory area where the chrominance component of the captured data is to be stored by CDACR2. In bothfield image capture, set the start address of the memory area where the chrominance component of the captured top-field image is to be stored by CDACR2. CDACR2 is not used in data fetch.

The address specified by this register must be in 32-bit units.

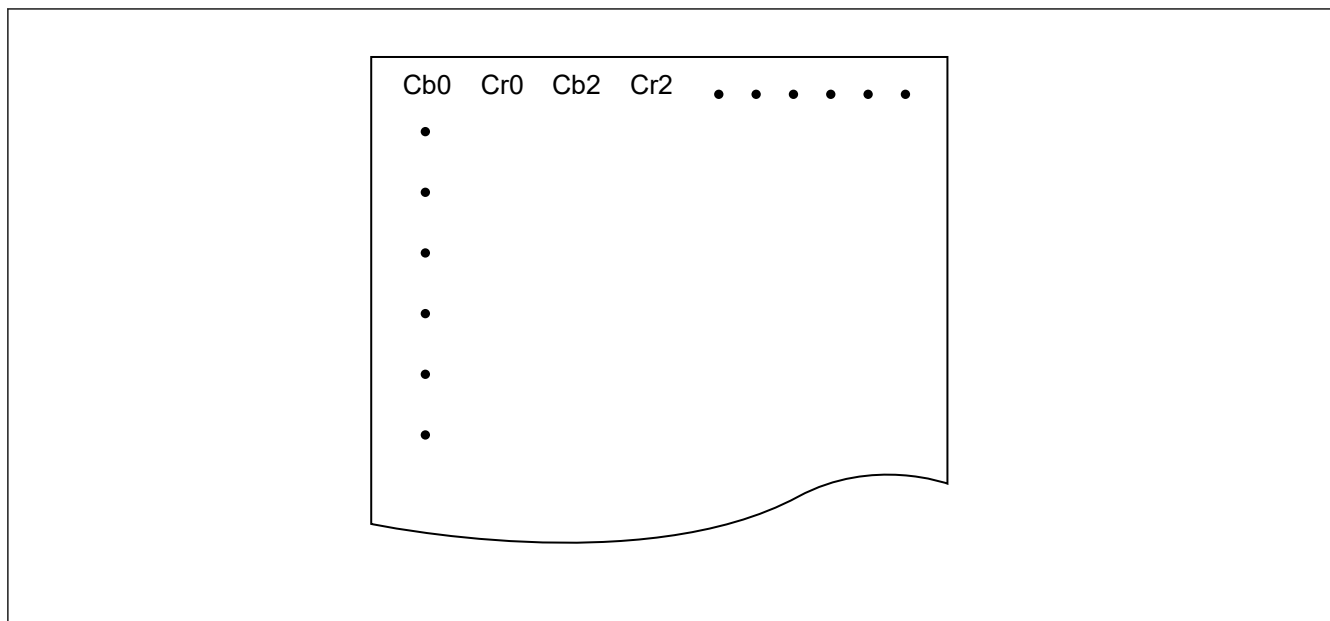
In this register, set the address of the starting point of the memory area where the captured data is to be stored in a bundle write, as shown in Figure 50.51.

- Frame image capture: Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored.
- One-field image capture: Set the address of the starting point of the memory area where the chrominance component of the captured image is to be stored.
- Both-field image capture: Set the address of the starting point of the memory area where the chrominance component of the captured top-field image is to be stored.

The chrominance component has an output data format as shown in [Figure 50.52](#), and is saved in the memory in this format.



**Figure 50.51** Relationship between captured image and chrominance component memory area

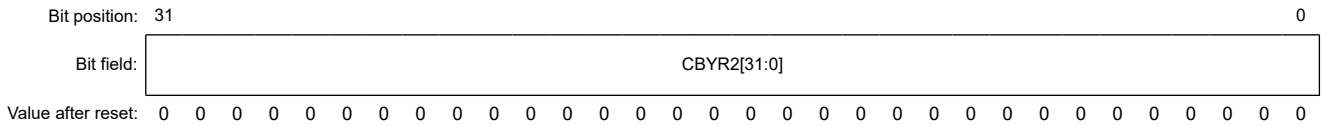


**Figure 50.52** Image of storing chrominance components in memory

### 50.2.27 CDBYR2, CDBYR2\_x : Capture Data Bottom-Field Address Y Register 2 (x = B, M)

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x0098 (CDBYR2)  
 0x1098 (CDBYR2\_B)  
 0x2098 (CDBYR2\_M)



Bit	Symbol	Function	R/W
31:0	CBYR2[31:0]	Set the address for storing the luminance component data of the captured bottom-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

Note: S-TYPE-3, P-TYPE-3

CDBYR2 specifies the address for the luminance component of the bottom field used in a bundle write. CDBYR2 is used only in a bundle write.

CDBYR2 specifies the address where the luminance component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the luminance component of the bottom-field image captured in both-field image capture is to be stored by CDBYR2. CDBYR2 is not used in frame image capture, one-field image capture, or data fetch.

The address specified by this register must be in 32-bit units.

In this register, set the address of the starting point of the memory area where the luminance component of the captured bottomfield image is to be stored in a bundle write, as shown in [Figure 50.53](#).

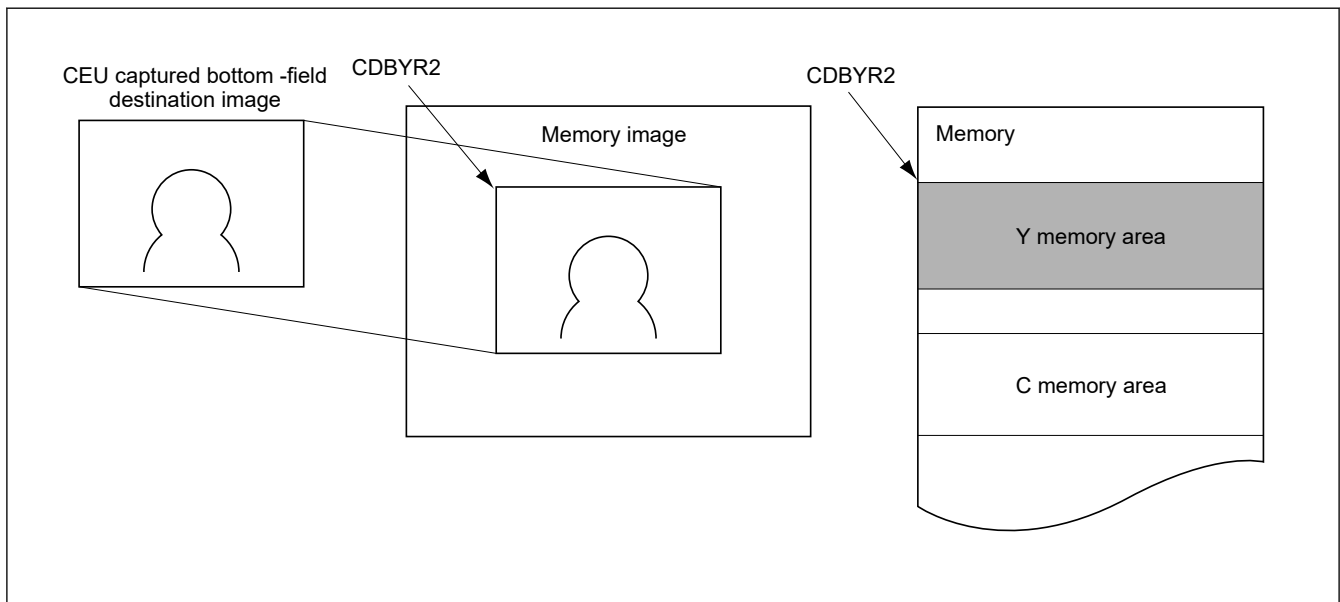
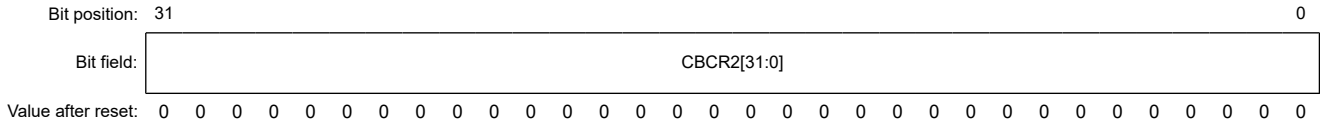


Figure 50.53 Relationship between captured bottom-field image and luminance component memory area

### 50.2.28 CDBCR2, CDBCR2\_x : Capture Data Bottom-Field Address C Register 2 (x = B, M)

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x009C (CDBCR2)  
 0x109C (CDBCR2\_B)  
 0x209C (CDBCR2\_M)



Bit	Symbol	Function	R/W
31:0	CBCR2[31:0]	Set the address for storing the chrominance component data of the captured bottom-field data (8-pixel units). The lower 3 bits should be written 0.	R/W

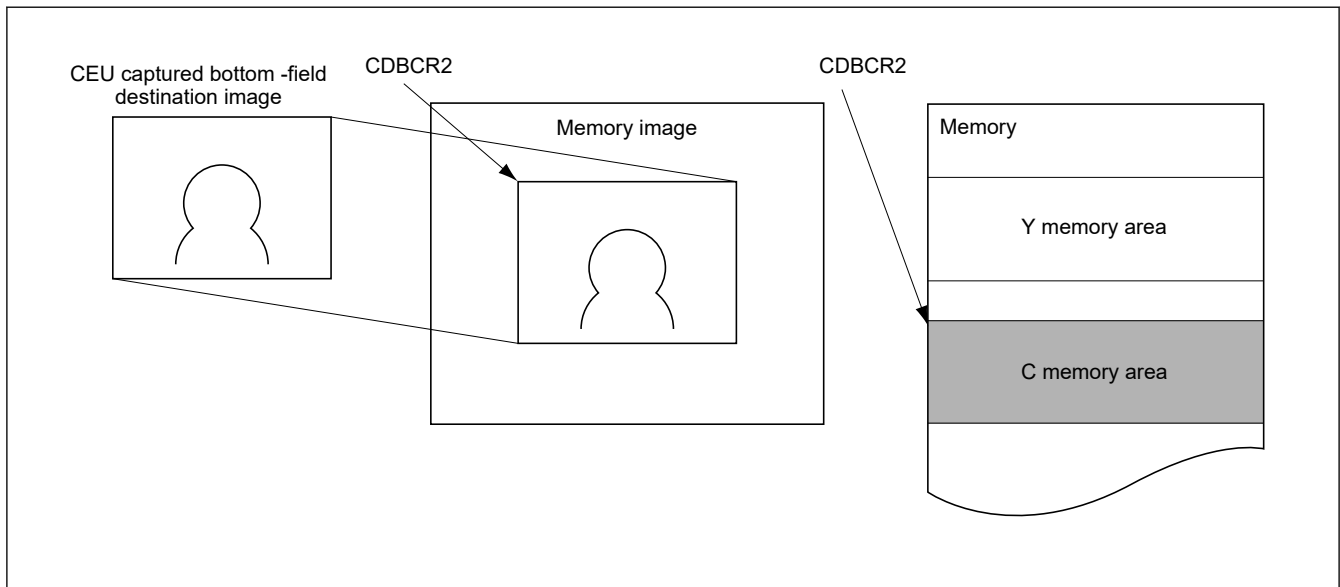
Note: S-TYPE-3, P-TYPE-3

CDBCR2 specifies the address for the chrominance component of the bottom field used in a bundle write. CDBCR2 is used only in a bundle write.

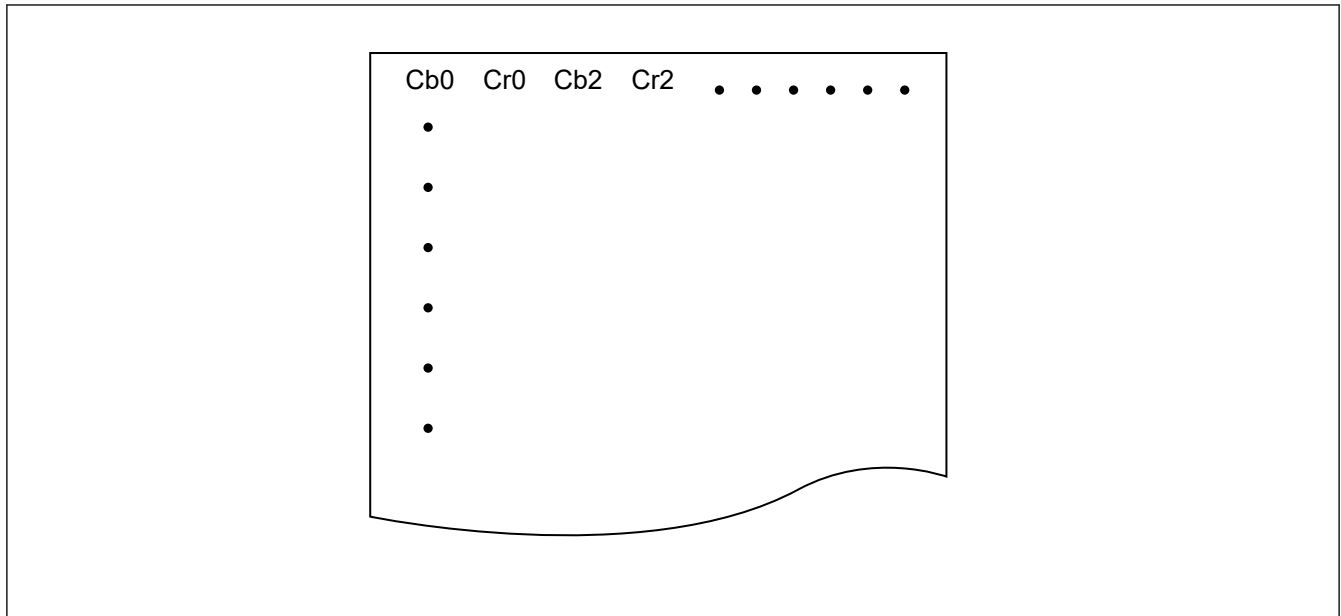
CDBCR2 specifies the address where the chrominance component of the captured bottom-field data is to be stored in both-field image capture. The CEU separates the captured image data into the luminance component data (Y) and the chrominance component data (C), and stores them in the memory via the bus. Set the start address of the memory area where the chrominance component of the bottom-field image captured in both-field image capture is to be stored by CDBCR2. CDBCR2 is not used in frame image capture, one-field image capture, or data fetch.

The address specified by this register must be in 32-bit units.

In this register, set the address of the starting point of the memory area where the chrominance component of the captured bottomfield image is to be stored in a bundle write, as shown in [Figure 50.54](#). The chrominance component has an output data format as shown in [Figure 50.55](#), and is saved in the memory in this format.



**Figure 50.54 Relationship between captured bottom-field image and chrominance component memory area**



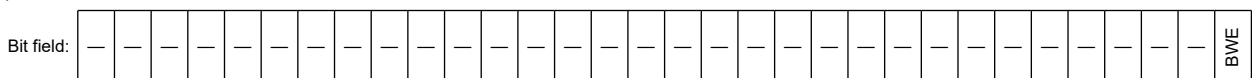
**Figure 50.55** Image of storing chrominance components in memory

### 50.2.29 CBWER : CEU Bufferable Write Enable Register

Base address: CEU = 0x4034\_8000  
 CEU\_NS = 0x5034\_8000

Offset address: 0x00A0

Bit position: 31



Value after reset: 0

Bit	Symbol	Function	R/W
0	BWE	Bufferable Write Enable 0: Disables Bufferable Write 1: Enables Bufferable Write	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: S-TYPE-3, P-TYPE-3

#### BWE bit (Bufferable Write Enable)

BWE bit indicates bufferable write is either enabled or disabled.

Note that if this bit is 1, even if the write access as CEU is completed, the actual slave write may not have ended.

Note: When this bit is 1, it requests an early response by writing write data to the temporary buffer if possible to the slave. Therefore, even if write access as CEU is completed, the actual slave write may not be finished. For details on slave groups that support bufferable writes, see [section 14.4.2. Operations When a Bus Error Occurs](#) and respective module sections.

## 50.3 Usage Notes

### 50.3.1 Settings for the Module-Stop Function

CEU operation can be disabled or enabled using Module Stop Control Register C (MSTPCRC). The CEU is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see [section 10, Low Power Modes](#).

### 50.3.2 Conditions for Connection to an External Module

#### (1) Clock Frequency

The external input clock (VIO\_CLK) should have a frequency at most the same as the CEU operating clock (PCLKA) frequency, with jitter on both sides included.

$$PCLKA \geq VIO\_CLK$$

#### (2) Blanking Period

The period from the last valid pixel in each line to the next horizontal sync signal HD must be at least 20 cycles.

#### (3) Fixed Period of Field Identification Signal

The field identification signal FLD should be fixed for at least 1-HD period since a VD input

### 50.3.3 Restrictions on Input/Output Functions

Table 50.11 lists the restrictions regarding the CEU input/output functions.

**Table 50.11 Restrictions on CEU Input/Output Functions**

Item	Restrictions
External module interface	The operating clock of the external module (VIO_CLK) should always have a frequency at most the same as that of the CEU operating (PCLKA), with jitter on both sides included.
	Selecting the interface, or modifying the frequency of the external module operating clock or HD/VD polarity must be done when capture operations are halted for sure.
	The capture horizontal size in image capture must be specified as 16-cycle units.
	The capture horizontal size in data fetch must be specified as 8-cycle units.
	The capture vertical size must be specified in 4-line units.
	The maximum number of cycles in the horizontal sync signal period should be 16,375 cycles of external input clock.
	The maximum number of lines (HD count) in the vertical sync signal should be 16,382 lines.
	The minimum number of captured pixels should be sub-QCIF (128 × 96).
	The maximum number of captured pixels should be 5 megapixels (2,560 × 1,920).
	The capture size in data enable fetch must be controlled by VD and HD in 4-byte units. The range of the capture size in data enable fetch that can be input is as follows. Maximum: 6 MB (2,048 × 1,536 × 2) Minimum: 16 bytes
Memory output	The output address must be specified in 32-bit units.
	The horizontal size of the destination image (memory) must be specified in 8-pixel units.
	The number of horizontal output pixels (= horizontal clipping size) must be specified in 8-pixel units.
	The number of vertical output lines (HD) (= vertical clipping size) must be specified in 4-line (HD) units.
	In data enable fetch bundle write, the output address must be specified in 32-byte units.
Internal processing	The filter clipping size must be specified as a value equal to or lower than the actual output size of the filter.



## 51. Internal Voltage Regulator

### 51.1 Overview

The MCU supports two power supply providing for core voltage (VDD):

- Switching regulator (DCDC)
- External power supply from external power device (External VDD)

Table 51.1 shows the specifications of Internal voltage regulator.

DCDC or External VDD is selected with the OFS setting.

- When OFS2.DCDCEN is 1, DCDC is selected (DCDC mode)
- When OFS2.DCDCEN is 0, External VDD is selected (External VDD mode)

**Table 51.1 Power supply selection**

OFS setting (OFS2.DCDCEN)	Power mode
1	DCDC mode
0	External VDD mode

### 51.2 Operation

#### 51.2.1 DCDC mode

Table 51.2 lists the DCDC mode pin settings, and Figure 51.1 shows the DCDC mode settings.

In DCDC mode, VDD is supplied from VCL through VLO output and an external inductor and capacitor.

**Table 51.2 DCDC mode pin settings**

Pins	Setting descriptions
All VCC and VCC2 pins	<ul style="list-style-type: none"> <li>• Connect each pin to the system power supply.</li> <li>• Connect each pin to VSS through a 0.1 <math>\mu</math>F multilayer ceramic capacitor. Place the capacitor close to the pin.</li> </ul>
VCC_DCDC pins	<ul style="list-style-type: none"> <li>• Connect the pin to the system power supply.</li> <li>• Connect the pin to VSS_DCDC through a 22 <math>\mu</math>F and 0.1-<math>\mu</math>F multilayer ceramic capacitor in parallel. Place the capacitor close to the pin.</li> </ul>
VCL pins	<ul style="list-style-type: none"> <li>• Connect each pin to VSS through a 0.22 <math>\mu</math>F multilayer ceramic capacitor. Place the capacitor close to the pin.</li> <li>• Connect each pin to an external inductor and capacitor. Place the inductor and capacitor close to the pin.(see Figure 51.1.)</li> </ul>
All VLO pins	Connect each pin to an external inductor and capacitor. Place the 2.2- $\mu$ H inductor and 47 $\mu$ F capacitor close to the pin. This capacitor must be grounded to VSS_DCDC

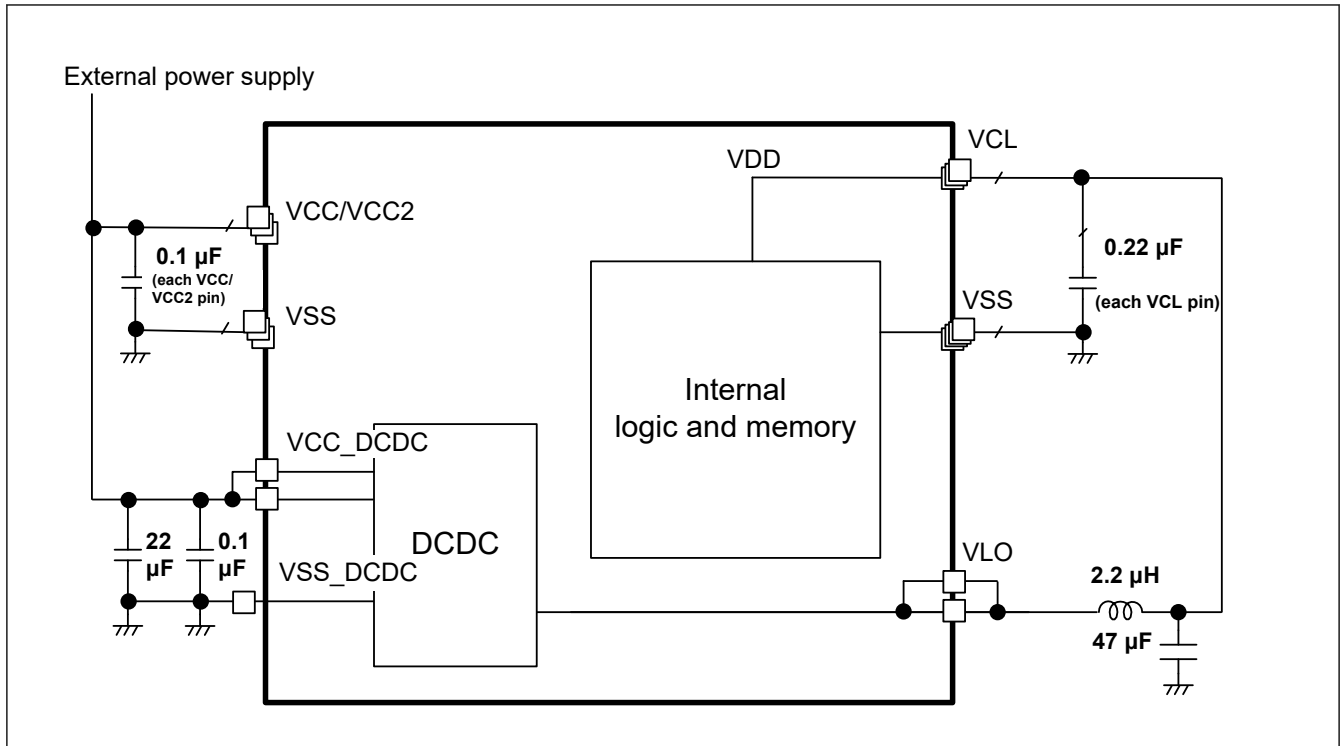


Figure 51.1 DCDC mode setting

### 51.2.2 External VDD mode

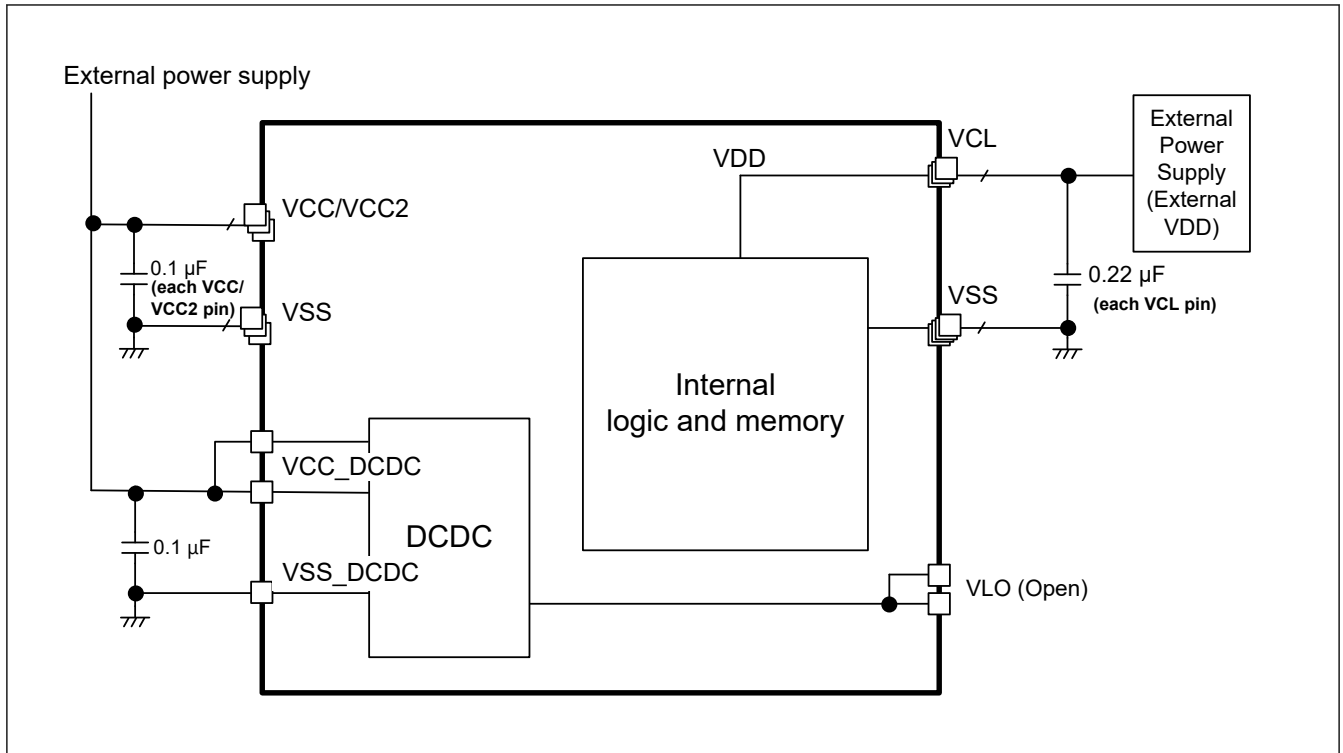
Table 51.3 lists the external VDD mode pin settings, and Figure 51.2 shows the external VDD mode settings.

VDD is supplied from VCL pins.

Note: Low-speed mode, Software Standby mode, Deep Software Standby mode 1, 2, 3 and Battery Backup Function are not supported in this mode.

Table 51.3 Setting Descriptions for external VDD mode

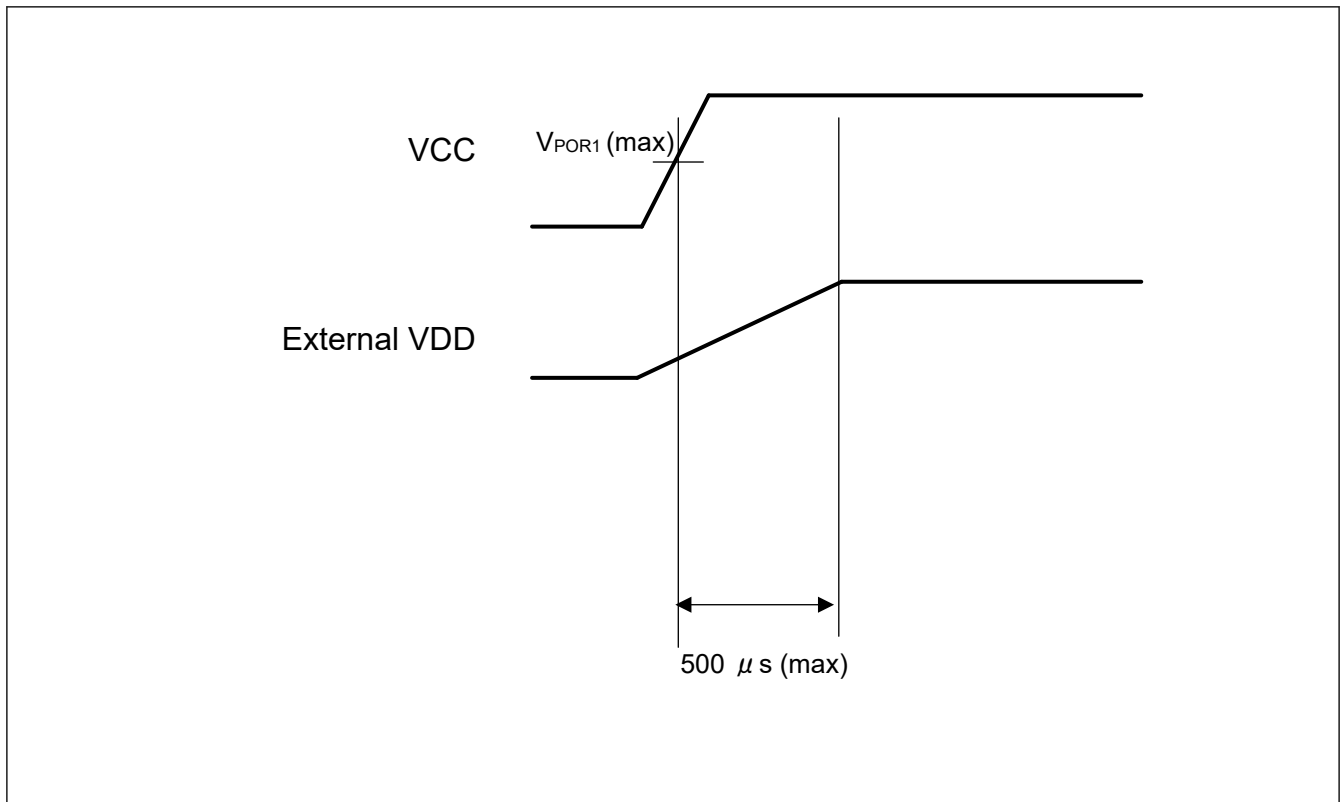
Pins	Setting descriptions
All VCC and VCC2 pins	<ul style="list-style-type: none"> <li>Connect each pin to the system power supply device for VCC.</li> <li>Connect each pin to VSS through a 0.1 µF multilayer ceramic capacitor. Place the capacitor close to the pin.</li> </ul>
VCC_DCDC pins	<ul style="list-style-type: none"> <li>Connect the pin to VSS_DCDC through 0.1 µF multilayer ceramic capacitor. Place the capacitor close to the pin.</li> </ul>
VCL pins	<ul style="list-style-type: none"> <li>Connect each pin to the system power supply device for VDD.</li> <li>Connect each pin to VSS through a 0.22 µF multilayer ceramic capacitor. Place the capacitor close to the pin.</li> </ul>
All VLO pins	Keep pin open



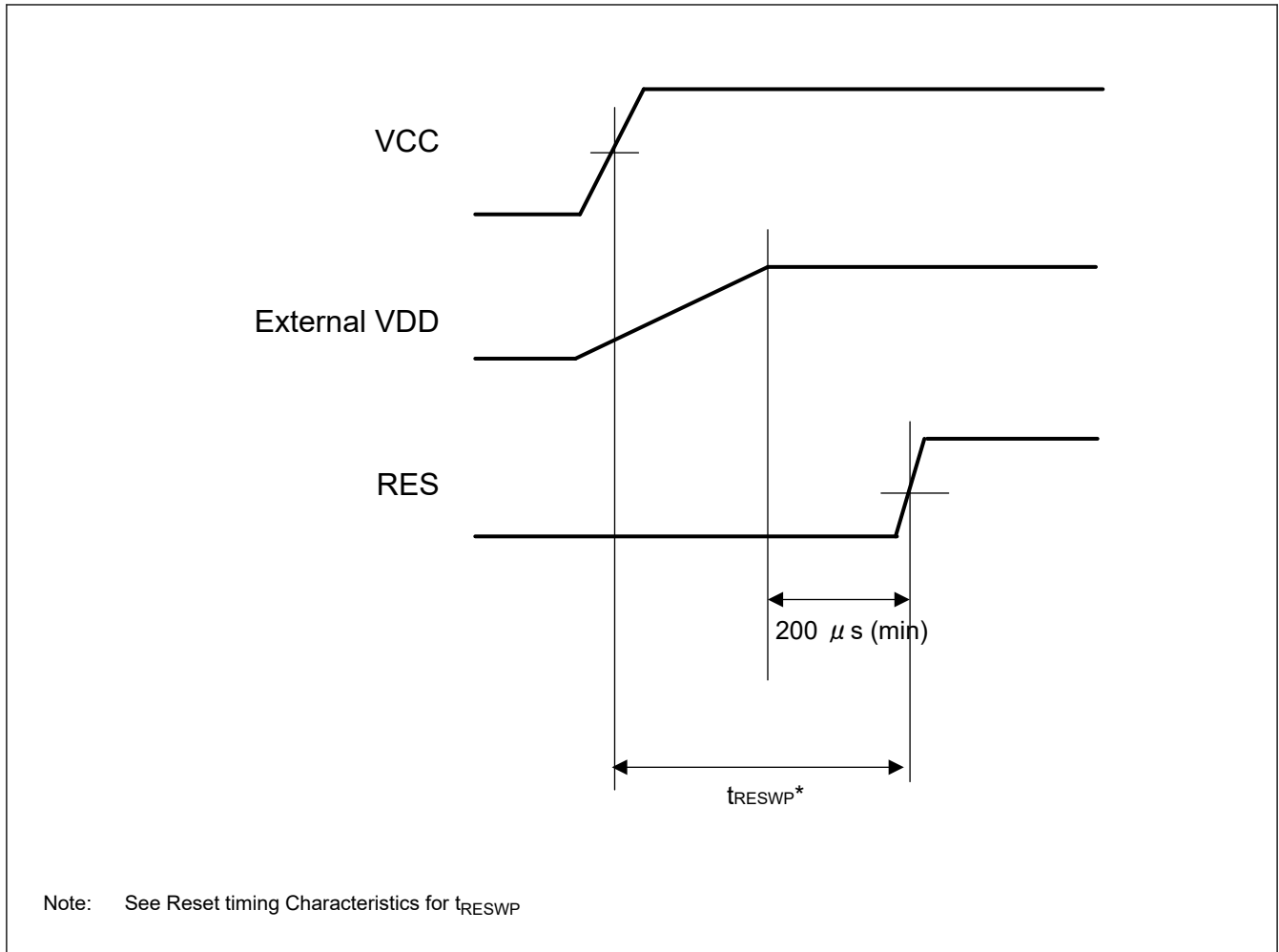
**Figure 51.2 External VDD mode setting**

In order to startup the device properly, use either of following procedure.

- VCL voltage must be raised 500  $\mu$ s after VCC voltage is raised to minimum VCC voltage.
- Raise VCC voltage with RES pin is low, and release RES pin after 200  $\mu$ s when VCL voltage is raised.



**Figure 51.3 Power-up sequence of external VDD mode without using RES pin**



**Figure 51.4** Power-up sequence of external VDD mode with using RES pin

### 51.3 Usage Notes

In DCDC mode, to minimize power loss and maximize efficiency, Renesas recommends using a 2.2- $\mu$ H inductor with a DC resistance of 100 m $\Omega$  or less. Short VCC and VCC\_DCDC.

In external VDD mode, the voltage of VDD should always be below the voltage of VCC including the power-on and power-off sequence. Refer to the chapter of electrical characteristics for more information.

## 52. Electrical Characteristics

Unless otherwise specified, minimum and maximum values are guaranteed by either design simulation, characterization results or test in production.

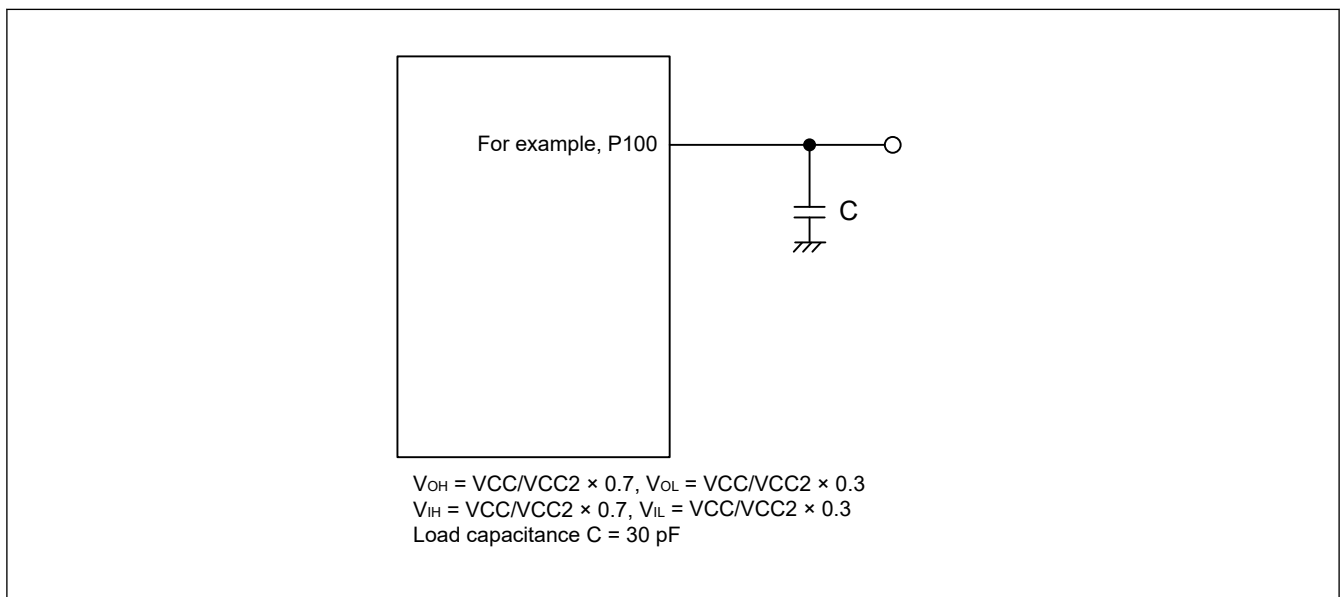
Supported peripheral functions and pins differ from one product name to another.

Unless otherwise specified, the electrical characteristics of the MCU are defined under the following conditions:

- $VCC = VCC\_DCDC = VCC\_USB = VBATT = 1.68$  to  $3.6$  V
- $VCC2 = 1.65$  to  $3.6$  V
- $AVCC0 = 1.65$  to  $3.6$  V
- $VREFH0 = 2.7$  V to  $AVCC0$
- $VREFH = 1.65$  V to  $AVCC0$
- $VSS = VSS\_DCDC = AVSS0 = VREFL0 / VREFL = VSS\_USB = 0$  V
- $VCC$  voltage is lower than  $2.7$  V :  $LVOCR.LVO0E = 1$ , otherwise  $LVOCR.LVO0E = 0$
- $VCC2$  voltage is lower than  $2.7$  V :  $LVOCR.LVO1E = 1$ , otherwise  $LVOCR.LVO1E = 0$
- $T_j = T_{opj}$

When not specified otherwise, typical values are measured at room temperature of  $25$  °C and  $VCC = VCC\_DCDC = VCC\_USB = VBATT = AVCC0 = VREFH0 = VREFH = 3.3$ V.

Figure 52.1 shows the timing conditions.



**Figure 52.1** Input or output timing measurement conditions

The recommended measurement conditions for the timing specification of each peripheral provided are for the best peripheral operation. Make sure to adjust the driving abilities of each pin to meet your conditions.

### 52.1 Absolute Maximum Ratings

**Table 52.1** Absolute maximum ratings (1 of 2)

Parameter	Symbol	Value	Unit
Power supply voltage	$VCC, VCC2, VCC\_DCDC, VCC\_USB^{*2}$	-0.3 to +4.0	V
External power supply voltage	VCL	-0.3 to +1.6	V
VBATT power supply voltage	VBATT	-0.3 to +4.0	V

**Table 52.1 Absolute maximum ratings (2 of 2)**

Parameter	Symbol	Value	Unit
Input voltage (except for 5 V-tolerant ports*1)	V <sub>in</sub>	-0.3 to VCC + 0.3 or -0.3 to VCC2 + 0.3	V
Input voltage (5 V-tolerant ports*1)	V <sub>in</sub>	-0.3 to + VCC + 4.0 (max. 5.8)	V
Reference power supply voltage	VREFH/VREFH0	-0.3 to AVCC0 + 0.3	V
Analog power supply voltage	AVCC0	-0.3 to +4.0	V
Analog input voltage	V <sub>AN</sub>	-0.3 to AVCC0 + 0.3	V
Operating junction temperature*3	T <sub>opj</sub>	-40 to +105	°C
Storage temperature	T <sub>stg</sub>	-55 to +125	°C

Note 1. Ports P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P715 and PB01 are 5 V tolerant.

Note 2. Connect VCC\_DCDC and VCC\_USB to VCC.

Note 3. See [section 52.2.1. Tj/Ta Definition](#).

**Caution: Permanent damage to the MCU might result if absolute maximum ratings are exceeded.**

**Table 52.2 Recommended operating conditions**

Parameter	Symbol	Min	Typ	Max	Unit	
Power supply voltages	VCC, VCC_DCDC	Other than the following	1.68	—	3.60	V
		When ETHERC/IIC Fast-mode+ is used	2.70	—	3.60	V
		When USB is used	3.00	—	3.60	V
	VCC2	1.65	—	3.60	V	
	VCL	When external VDD is used*2	1.20	—	1.25	V
		When DCDC is used (High-speed mode)	—	1.21	—	V
		When DCDC is used (Low-speed mode or Software Standby mode)	—	1.18	—	V
VSS, VSS_DCDC	—	0	—	V		
USB power supply voltages	VCC_USB	—	VCC	—	V	
	VSS_USB	—	0	—	V	
VBATT power supply voltage	VBATT	1.62	—	3.60	V	
Analog power supply voltages	AVCC0*1	When ADC is not used	1.65	—	3.60	V
		When ADC is used	2.70	—	3.60	V
	AVSS0	—	0	—	V	

Note 1. When the A/D converter, the D/A converter and the High-Speed Analog Comparator are not in use, do not leave the AVCC0, VREFH/VREFH0, AVSS0, and VREFL/VREFL0 pins open. Connect the AVCC0 and VREFH/VREFH0 pins to VCC, and the AVSS0 and VREFL/VREFL0 pins to VSS, respectively.

Note 2. VCL voltage must never be higher than VCC voltage.

## 52.2 DC Characteristics

### 52.2.1 Tj/Ta Definition

**Table 52.3 DC characteristics**

Parameter	Symbol	Typ	Max	Unit	Test conditions
Permissible operating junction temperature	T <sub>j</sub>	—	105	°C	High-speed mode Low-speed mode

Note: Make sure that  $T_j = T_a + \theta_{ja} \times \text{total power consumption (W)}$ , where total power consumption =  $(VCC - V_{OH}) \times \Sigma I_{OH} + V_{OL} \times \Sigma I_{OL} + (I_{CCmax} + I_{CC\_DCDCmax}) \times VCC$ .

Note: Minimum Ambient Temperature(Ta) is -40°C

## 52.2.2 I/O $V_{IH}$ , $V_{IL}$

**Table 52.4 I/O  $V_{IH}$ ,  $V_{IL}$  except for Schmitt trigger input pins**

Parameter		VCC/VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit
Peripheral function pins	EXTAL (external clock input), WAIT, SPI* <sup>1</sup> (except RSPCK)	1.68 V or above	$V_{IH}$	$VCC \times 0.8$	—	—	V
			$V_{IL}$	—	—	$VCC \times 0.2$	
	SPI* <sup>2</sup> (except RSPCKB_A)	1.65 V or above	$V_{IH}$	$VCC2 \times 0.8$	—	—	
			$V_{IL}$	—	—	$VCC2 \times 0.2$	
	OSPI (except OM_RSTO1 and OM_ECSINT1)	2.70 V or above	$V_{IH}$	$VCC2 \times 0.8$	—	—	
			$V_{IL}$	—	—	$VCC2 \times 0.2$	
		1.65 V or above	$V_{IH}$	$VCC2 \times 0.7$	—	$VCC2 + 0.3$	
			$V_{IL}$	$VSS - 0.3$	—	$VCC2 \times 0.3$	
	TMS, TDI, TCK, SWDIO, SWCLK	1.68 V or above	$V_{IH}$	$VCC \times 0.7$	—	—	
			$V_{IL}$	—	—	$VCC \times 0.3$	
	ETHERC	2.70 V or above	$V_{IH}$	2.3	—	—	
			$V_{IL}$	—	—	$VCC \times 0.2$	
	IIC (SMBus)	2.70 V or above	$V_{IH}$	2.1	—	$VCC + 3.6$ (max 5.8)	
			$V_{IL}$	—	—	0.8	
RTCIC0, RTCIC1, RTCIC2, EXCIN when VCC power supply is selected	1.68 V or above	$V_{IH}$	0.9	—	3.9		
		$V_{IL}$	—	—	0.3		
RTCIC0, RTCIC1, RTCIC2, EXCIN when VBATT power supply is selected	1.68 V or above	$V_{IH}$	0.9	—	3.9		
		$V_{IL}$	—	—	0.3		

Note 1. SPI0\_A, SPI0\_B, SPI0\_C and SPI1\_B

Note 2. SPI1\_A

Note 3. RES and peripheral function pins associated with P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P713 (total 22 pins).

Note 4. All input pins except for the peripheral function pins already described in the table. There is an item for each power supply voltage for each port. Refer to the IO chapter for the power supply of the port.

Note 5. P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P713 (total 22 pins).

Note 6. All input pins except for the ports already described in the table. There is an item for each power supply voltage for each port. Refer to the IO chapter for the power supply of the port.

Note 7. When VCC is less than 1.68 V, the input voltage of 5 V-tolerant ports should be less than 3.6 V, otherwise breakdown may occur because 5 V-tolerant ports are electrically controlled so as not to violate the break down voltage.

Table 52.5 I/O  $V_{IH}$ ,  $V_{IL}$  of Schmitt trigger input pins

Parameter		VCC/VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit
Peripheral function pins	IIC (except for SMBus)	1.68 V or above	$V_{IH}$	$VCC \times 0.7$	—	$VCC + 3.6$ (max 5.8)	V
			$V_{IL}$	—	—	$VCC \times 0.3$	
			$\Delta V_T$	$VCC \times 0.05$	—	—	
	5 V-tolerant ports <sup>*3*7</sup>	1.68 V or above	$V_{IH}$	$VCC \times 0.8$	—	$VCC + 3.6$ (max 5.8)	
			$V_{IL}$	—	—	$VCC \times 0.2$	
			$\Delta V_T$	$VCC \times 0.05$	—	—	
	Other VCC input pins <sup>*4</sup>	1.68 V or above	$V_{IH}$	$VCC \times 0.8$	—	—	
			$V_{IL}$	—	—	$VCC \times 0.2$	
			$\Delta V_T$	$VCC \times 0.05$	—	—	
	Other VCC2 input pins <sup>*4</sup>	1.65 V or above	$V_{IH}$	$VCC2 \times 0.8$	—	—	
			$V_{IL}$	—	—	$VCC2 \times 0.2$	
			$\Delta V_T$	$VCC2 \times 0.05$	—	—	
Other AVCC0 input pins <sup>*4</sup>	1.65 V or above	$V_{IH}$	$AVCC0 \times 0.8$	—	—		
		$V_{IL}$	—	—	$AVCC0 \times 0.2$		
		$\Delta V_T$	$AVCC0 \times 0.05$	—	—		
Ports	5 V-tolerant port <sup>*5*7</sup>	1.68 V or above	$V_{IH}$	$VCC \times 0.8$	—	$VCC + 3.6$ (max 5.8)	V
			$V_{IL}$	—	—	$VCC \times 0.2$	
	Other VCC input pins <sup>*6</sup>	1.68 V or above	$V_{IH}$	$VCC \times 0.8$	—	—	
			$V_{IL}$	—	—	$VCC \times 0.2$	
	Other VCC2 input pins <sup>*6</sup>	1.65 V or above	$V_{IH}$	$VCC2 \times 0.8$	—	—	
			$V_{IL}$	—	—	$VCC2 \times 0.2$	
	Other AVCC0 input pins <sup>*6</sup>	1.65 V or above	$V_{IH}$	$AVCC0 \times 0.8$	—	—	
			$V_{IL}$	—	—	$AVCC0 \times 0.2$	

Note 1. SPI0\_A, SPI0\_B, SPI0\_C and SPI1\_B

Note 2. SPI1\_A

Note 3. RES and peripheral function pins associated with P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P713 (total 22 pins).

Note 4. All input pins except for the peripheral function pins already described in the table. There is an item for each power supply voltage for each port. Refer to the IO chapter for the power supply of the port.

Note 5. P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P713 (total 22 pins).

Note 6. All input pins except for the ports already described in the table. There is an item for each power supply voltage for each port. Refer to the IO chapter for the power supply of the port.

Note 7. When VCC is less than 1.68 V, the input voltage of 5 V-tolerant ports should be less than 3.6 V, otherwise breakdown may occur because 5 V-tolerant ports are electrically controlled so as not to violate the break down voltage.



52.2.3 I/O  $I_{OH}$ ,  $I_{OL}$ Table 52.6 I/O  $I_{OH}$ ,  $I_{OL}$  (1 of 2)

Parameter			VCC/ VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit		
Permissible output current (average value per pin)	Ports P000 to P010, P014, P015, P201	—	—	$I_{OH}$	—	—	-2.0	mA		
				$I_{OL}$	—	—	2.0	mA		
	Ports P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P713 (total 22 pins)	Low drive*1	—	—	$I_{OH}$	—	—	-2.0	mA	
					$I_{OL}$	—	—	2.0	mA	
		Middle drive*2	—	—	—	$I_{OH}$	—	—	-4.0	mA
						$I_{OL}$	—	—	4.0	mA
		High drive*3	—	—	—	$I_{OH}$	—	—	-20	mA
						$I_{OL}$	—	—	20.0	mA
	Ports P100 to P103, P304 to P308, P800 to P804, P808 to P809 (total 16 pins)	Low drive*1	—	—	—	—	—	-2.0	mA	
									$I_{OL}$	—
		Middle drive*2	—	—	—	—	—	—	-4.0	mA
									$I_{OL}$	—
		High drive*3	—	—	—	—	—	—	-16	mA
									$I_{OL}$	—
		High-speed high drive*4	—	—	—	—	—	—	-20	mA
									$I_{OL}$	—
	Other output pins*5	Low drive*1	—	—	—	—	—	-2.0	mA	
									$I_{OL}$	—
		Middle drive*2	—	—	—	—	—	—	-4.0	mA
									$I_{OL}$	—
High drive*3		—	—	—	—	—	—	-16	mA	
								$I_{OL}$	—	—

Table 52.6 I/O  $I_{OH}$ ,  $I_{OL}$  (2 of 2)

Parameter			VCC/ VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit	
Permissible output current (max value per pin)	Ports P000 to P010, P014, P015, P201	—	—	$I_{OH}$	—	—	-4.0	mA	
				$I_{OL}$	—	—	4.0	mA	
	Ports P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P713 (total 22 pins)	Low drive* <sup>1</sup>	—	—	$I_{OH}$	—	—	-4.0	mA
					$I_{OL}$	—	—	4.0	mA
		Middle drive* <sup>2</sup>	—	—	$I_{OH}$	—	—	-8.0	mA
					$I_{OL}$	—	—	8.0	mA
		High drive* <sup>3</sup>	—	—	$I_{OH}$	—	—	-40	mA
					$I_{OL}$	—	—	40.0	mA
	Ports P100 to P103, P304 to P308, P800 to P804, P808 to P809 (total 16 pins)	Low drive* <sup>1</sup>	—	—	$I_{OH}$	—	—	-4.0	mA
					$I_{OL}$	—	—	4.0	mA
		Middle drive* <sup>2</sup>	—	—	$I_{OH}$	—	—	-8.0	mA
					$I_{OL}$	—	—	8.0	mA
		High drive* <sup>3</sup>	—	—	$I_{OH}$	—	—	-32	mA
					$I_{OL}$	—	—	32.0	mA
		High-speed high drive* <sup>4</sup>	—	—	$I_{OH}$	—	—	-40	mA
					$I_{OL}$	—	—	40.0	mA
	Other output pins* <sup>5</sup>	Low drive* <sup>1</sup>	—	—	$I_{OH}$	—	—	-4.0	mA
					$I_{OL}$	—	—	4.0	mA
		Middle drive* <sup>2</sup>	—	—	$I_{OH}$	—	—	-8.0	mA
					$I_{OL}$	—	—	8.0	mA
High drive* <sup>3</sup>		—	—	$I_{OH}$	—	—	-32	mA	
				$I_{OL}$	—	—	32.0	mA	
Permissible output current (max value of total of all pins)	Maximum of all output pins	VCC I/O	1.68 V or above	$\Sigma I_{OH} (max)$	—	—	-80	mA	
		VCC2 I/O	1.65 V or above		—	—	-80		
		AVCC0 I/O	1.65 V or above		—	—	-33		
		VCC and VCC2 I/O	1.65 V or above	$\Sigma I_{OL} (max)$	—	—	80	mA	
		AVCC0 I/O	1.65 V or above		—	—	33		

Note 1. This is the value when low driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability except for 400 and P401 is retained in Deep Software Standby mode.

Note 2. This is the value when middle driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability except for 400 and P401 is retained in Deep Software Standby mode.

Note 3. This is the value when high driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability except for 400 and P401 is retained in Deep Software Standby mode.

Note 4. This is the value when high-speed high driving ability is selected in the Port Drive Capability in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.

Note 5. Except for P200, which is an input port.

**Caution:** To protect the reliability of the MCU, the output current values should not exceed the values in this table. The average output current indicates the average value of current measured during 100  $\mu$ s.

52.2.4 I/O  $V_{OH}$ ,  $V_{OL}$ , and Other CharacteristicsTable 52.7 I/O  $V_{OH}$ ,  $V_{OL}$ , and other characteristics (1 of 2)

Parameter		VCC/VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit	Test conditions
Output voltage	IIC	2.70 V or above	$V_{OL}$	—	—	0.4	V	$I_{OL} = 3.0 \text{ mA}$
			$V_{OL}$	—	—	0.6		$I_{OL} = 6.0 \text{ mA}$
		1.68 V or above	$V_{OL}$	—	—	$VCC \times 0.2$		$I_{OL} = 3.0 \text{ mA}$
			$V_{OL}$	—	—	0.6		$I_{OL} = 6.0 \text{ mA}$
	IIC*1	2.70 V or above	$V_{OL}$	—	—	0.4		$I_{OL} = 15.0 \text{ mA}$ (ICFER.FMPE = 1)
			$V_{OL}$	—	0.4	—		$I_{OL} = 20.0 \text{ mA}$ (ICFER.FMPE = 1)
	ETHERC	2.70 V or above	$V_{OH}$	$VCC - 0.5$	—	—		$I_{OH} = -1.0 \text{ mA}$
			$V_{OL}$	—	—	0.4		$I_{OL} = 1.0 \text{ mA}$
	Ports P205, P206, P402 to P404, P406 to P415, P511, P512, P709 to P713 (total 22 pins)*2	—	$V_{OH}$	$VCC - 1.0$	—	—		$I_{OH} = -20 \text{ mA}$ $VCC = 3.3 \text{ V}$
		—	$V_{OL}$	—	—	1		$I_{OL} = 20 \text{ mA}$ $VCC = 3.3 \text{ V}$
	Other output pins	1.68 V or above	$V_{OH}$	$VCC - 0.5$	—	—		$I_{OH} = -1.0 \text{ mA}$
			$V_{OL}$	—	—	0.5		$I_{OL} = 1.0 \text{ mA}$
		1.65V or above	$V_{OH}$	$VCC2 - 0.5$	—	—		$I_{OH} = -1.0 \text{ mA}$
			$V_{OL}$	—	—	0.5		$I_{OL} = 1.0 \text{ mA}$
			$V_{OH}$	$AVCC0 - 0.5$	—	—		$I_{OH} = -1.0 \text{ mA}$
$V_{OL}$			—	—	0.5	$I_{OL} = 1.0 \text{ mA}$		
Input leakage current	RES	1.68 V or above	$ I_{in} $	—	—	5.0	$\mu\text{A}$	$V_{in} = 0 \text{ V}$ $V_{in} = 5.5 \text{ V}$
	Port P200			—	—	1.0		$V_{in} = 0 \text{ V}$ $V_{in} = VCC$
Three-state leakage current (off state)	5 V-tolerant ports	1.68 V or above	$ I_{TSI} $	—	—	5.0	$\mu\text{A}$	$V_{in} = 0 \text{ V}$ $V_{in} = 5.5 \text{ V}$
	Other ports (except for port P200)	1.68 V or above		—	—	1.0		$V_{in} = 0 \text{ V}$ $V_{in} = VCC$
		1.65 V or above		—	—	1.0		$V_{in} = 0 \text{ V}$ $V_{in} = VCC2, AVCC0$
Input pull-up MOS current	Ports P0 to P9	2.70 V or above	$I_p$	-300	—	-10	$\mu\text{A}$	$VCC, VCC2, AVCC0 = 2.7 \text{ to } 3.6 \text{ V}$ $V_{in} = 0 \text{ V}$
		1.68 V or above		-300	—	-5		$VCC = 1.68 \text{ to } 3.6 \text{ V}$ $V_{in} = 0 \text{ V}$
		1.65 V or above		-300	—	-5		$VCC2, AVCC0 = 1.65 \text{ to } 3.6 \text{ V}$ $V_{in} = 0 \text{ V}$

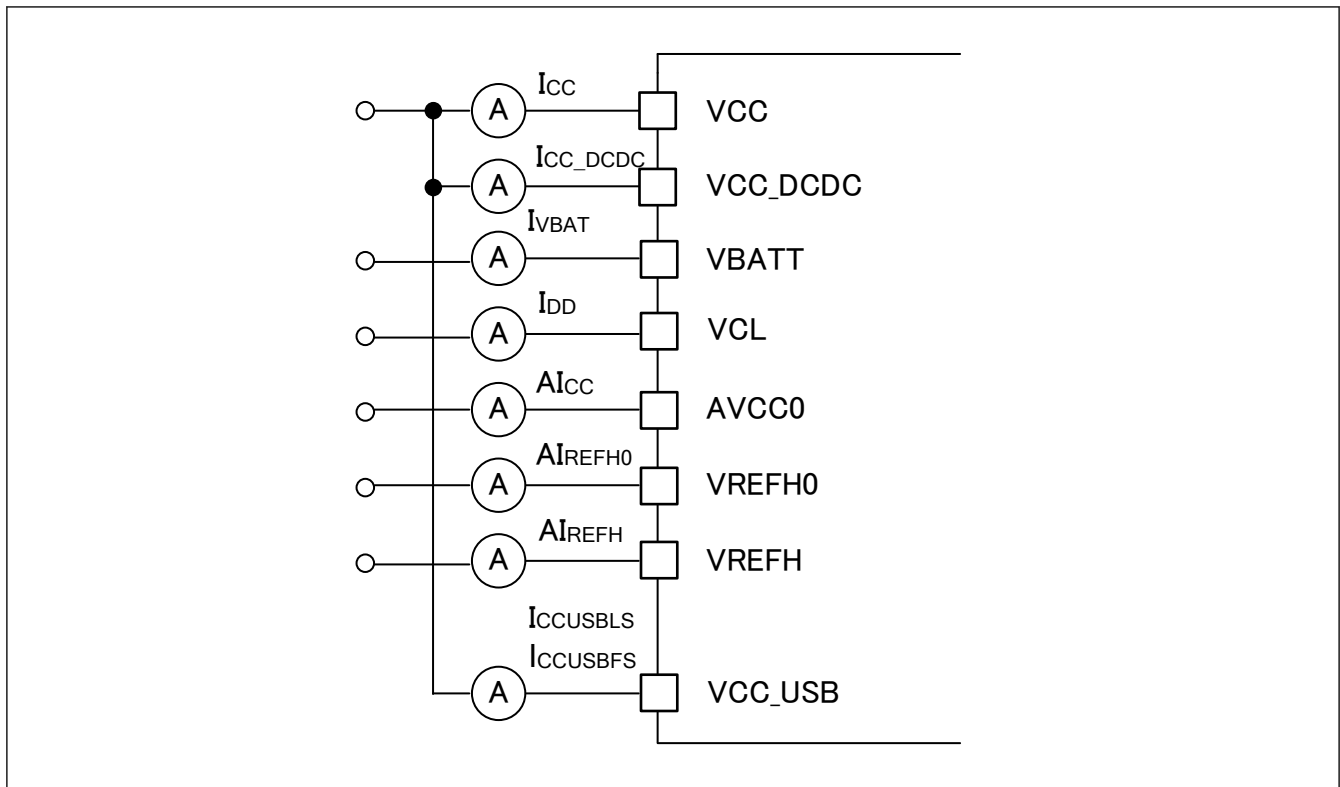
**Table 52.7 I/O  $V_{OH}$ ,  $V_{OL}$ , and other characteristics (2 of 2)**

Parameter		VCC/VCC2/ AVCC0	Symbol	Min	Typ	Max	Unit	Test conditions
Input capacitance	Ports P014, P015	—	$C_{in}$	—	—	16	pF	Vbias = 0 V Vamp = 20 mV f = 1 MHz Ta = 25°C
	Ports P814/ USB_DP, P815/ USB_DM	—		—	—	12		
	Ports P400, P401, P409, P410, P511, P512	—		—	—	10		
	Other input pins	—		—	—	8		

Note 1. SCL0\_A, SDA0\_A, SCL1\_A, SDA1\_A (total 4 pins).

Note 2. This is the value when high driving ability is selected in the Port Drive Capability bit in the PmnPFS register. The selected driving ability is retained in Deep Software Standby mode.

### 52.2.5 Operating and Standby Current



**Figure 52.2 Consumption current measurement diagram**

**Table 52.8 Current of high-speed mode, maximum condition (MVE and peripheral operation) (DCDC mode)**

Parameter		Symbol	Typ	Max	Unit	Test conditions		
Supply current <sup>*1*2</sup>	—	I <sub>CC</sub>	2.8	7.05	mA			
	CPUCLK = 360 MHz	VCC_DCDC ≥ 2.5 V	I <sub>CC_DCDC</sub> <sup>*5</sup>	122	243	mA	VCC_DCDC = 3.3 V CPUCLK = 360 MHz, ICLK = 120 MHz, PCLKA = 120 MHz, PCLKB = 60 MHz, PCLKC = 60 MHz, PCLKD = 120 MHz, PCLKE = 120 MHz, FCLK = 60 MHz	
			I <sub>DD</sub> <sup>*3</sup>	287	501			
		VCC_DCDC < 2.5 V	I <sub>CC_DCDC</sub> <sup>*5</sup>	224	320			VCC_DCDC = 1.8 V
			I <sub>DD</sub>	287	400 <sup>*4</sup>			
	CPUCLK = 240 MHz	VCC_DCDC ≥ 2.5 V	I <sub>CC_DCDC</sub> <sup>*5</sup>	95	210	mA	VCC_DCDC = 3.3 V CPUCLK = 240 MHz, ICLK = 240 MHz, PCLKA = 120 MHz, PCLKB = 60 MHz, PCLKC = 60 MHz, PCLKD = 120 MHz, PCLKE = 120 MHz, FCLK = 60 MHz	
			I <sub>DD</sub> <sup>*3</sup>	224	432			
		VCC_DCDC < 2.5 V	I <sub>CC_DCDC</sub> <sup>*5</sup>	175	320			VCC_DCDC = 1.8 V
I <sub>DD</sub>			224	400 <sup>*4</sup>				

- Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
- Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
- Note 3. I<sub>DD</sub> depends on f (CPUCLK and ICLK) as follows.  
I<sub>DD</sub> Max. = 0.68 × f CPUCLK + 0.41 × f ICLK + 175 (unit : mA, fCPUCLK and fICLK are MHz)
- Note 4. Do not actual consumption current during operation exceed the current value described here in VCC\_DCDC < 2.5V.
- Note 5. Typical DCDC efficiency is applied.

**Table 52.9 Current of high-speed mode, maximum condition (MVE and peripheral operation) (External VDD mode)**

Parameter	CPUCLK Frequency	Symbol	Typ	Max	Unit	Test conditions
Supply current <sup>*1*2</sup>	—	I <sub>CC</sub>	2.8	7.05	mA	
	CPUCLK = 360 MHz	I <sub>DD</sub> <sup>*3</sup>	287	501	mA	CPUCLK = 360 MHz, ICLK = 120 MHz, PCLKA = 120 MHz, PCLKB = 60 MHz, PCLKC = 60 MHz, PCLKD = 120 MHz, PCLKE = 120 MHz, FCLK = 60 MHz
	CPUCLK = 240 MHz	I <sub>DD</sub> <sup>*3</sup>	224	432	mA	CPUCLK = 240 MHz, ICLK = 240 MHz, PCLKA = 120 MHz, PCLKB = 60 MHz, PCLKC = 60 MHz, PCLKD = 120 MHz, PCLKE = 120 MHz, FCLK = 60 MHz

- Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
- Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
- Note 3. I<sub>DD</sub> depends on f (CPUCLK and ICLK) as follows.  
I<sub>DD</sub> Max. = 0.68 × f CPUCLK + 0.41 × f ICLK + 175 (unit : mA, fCPUCLK and fICLK are MHz)

**Table 52.10 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock ON (DCDC mode)**

Parameter		Symbol	Typ	Max	Unit	Test conditions
Supply current <sup>*1*2</sup>	CPUCLK = 360 MHz	I <sub>CC_DCDC</sub> <sup>*4</sup>	119	228	mA	VCC_DCDC = 3.3 V <sup>*5</sup>
		I <sub>DD</sub> <sup>*3</sup>	279	469		
	CPUCLK = 240 MHz	I <sub>CC_DCDC</sub> <sup>*4</sup>	92	194		
		I <sub>DD</sub> <sup>*3</sup>	215	399		

- Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
- Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.

- Note 3.  $I_{DD}$  depends on  $f$  (CPUCLK and ICLK) as follows.  
 $I_{DD} \text{ Max.} = 0.67 \times f \text{ CPUCLK} + 0.29 \times f \text{ ICLK} + 175$  (unit : mA, fCPUCLK and fICLK are MHz)
- Note 4. Typical DCDC efficiency is applied.
- Note 5. Same frequency condition is applied as in the maximum condition.

**Table 52.11 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock ON (External VDD mode)**

Parameter	Symbol	Typ	Max	Unit	Test conditions
Supply current <sup>*1*2</sup>	CPUCLK = 360 MHz	$I_{DD}^{*3}$	279	mA	*4
	CPUCLK = 240 MHz	$I_{DD}^{*3}$	215		

- Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
- Note 2. Measured with clocks supplied to the peripheral functions and peripherals being operated. This does not include the BGO operation.
- Note 3.  $I_{DD}$  depends on  $f$  (CPUCLK and ICLK) as follows.  
 $I_{DD} \text{ Max.} = 0.67 \times f \text{ CPUCLK} + 0.29 \times f \text{ ICLK} + 175$  (unit : mA, fCPUCLK and fICLK are MHz)
- Note 4. Same frequency condition is applied as in the maximum condition.

**Table 52.12 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock OFF (DCDC mode)**

Parameter	Symbol	Typ	Max	Unit	Test conditions
Supply current <sup>*1*2</sup>	CPUCLK = 360 MHz	$I_{CC\_DCDC}^{*4}$	111	mA	VCC_DCDC = 3.3 V *5
		$I_{DD}^{*3}$	261		
	CPUCLK = 240 MHz	$I_{CC\_DCDC}^{*4}$	83		
		$I_{DD}^{*3}$	194		

- Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
- Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
- Note 3.  $I_{DD}$  depends on  $f$  (CPUCLK and ICLK) as follows.  
 $I_{DD} \text{ Max.} = 0.68 \times f \text{ CPUCLK} + 0.17 \times f \text{ ICLK} + 175$  (unit : mA, fCPUCLK and fICLK are MHz)
- Note 4. Typical DCDC efficiency is applied.
- Note 5. Same frequency condition is applied as in the maximum condition.

**Table 52.13 Current of high-speed mode, maximum data processing (MVE operation), peripheral clock OFF (External VDD mode)**

Parameter	Symbol	Typ	Max	Unit	Test conditions
Supply current <sup>*1*2</sup>	CPUCLK = 360 MHz	$I_{DD}^{*3}$	261	mA	*4
	CPUCLK = 240 MHz	$I_{DD}^{*3}$	194		

- Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
- Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
- Note 3.  $I_{DD}$  depends on  $f$  (CPUCLK and ICLK) as follows.  
 $I_{DD} \text{ Max.} = 0.68 \times f \text{ CPUCLK} + 0.17 \times f \text{ ICLK} + 175$  (unit : mA, fCPUCLK and fICLK are MHz)
- Note 4. Same frequency condition is applied as in the maximum condition.

**Table 52.14 Current of high-speed mode, CPU Sleep mode (DCDC mode and External VDD mode)**

Parameter	Symbol	Typ	Max	Unit	Test conditions
Supply current <sup>*1*3*4</sup>	CPUCLK = 240 MHz	$I_{DD}^{*2}$	29	mA	—

- Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.
- Note 2.  $I_{DD}$  depends on  $f$  (CPUCLK and ICLK) as follows.  
 $I_{DD} \text{ Typ.} = 0.063 \times f \text{ CPUCLK} + 0.13 \times f \text{ ICLK} + 17.6$  (unit : mA, fCPUCLK and fICLK are MHz)  
 $I_{DD} \text{ Max.} = 0.063 \times f \text{ CPUCLK} + 0.13 \times f \text{ ICLK} + 175$  (unit : mA, fCPUCLK and fICLK are MHz)
- Note 3. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.
- Note 4. ICLK, FCLK, PCLKA, PCLKB, PCLKC, PCLKD and PCLK E are set to divided by 64.

**Table 52.15 Current of high-speed mode, CPU Deep Sleep mode (DCDC mode and External VDD mode)**

Parameter		Symbol	Typ	Max	Unit	Test conditions
Supply current <sup>*1*3*4</sup>	CPUCLK = 240 MHz	$I_{DD}^{*2}$	12	85	mA	—

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Note 2.  $I_{DD}$  depends on f (ICLK) as follows.

$I_{DD}$  Typ. =  $0.13 \times f_{ICLK} + 5$  (unit : mA, fCPUCLK and fICLK are MHz)

$I_{DD}$  Max. =  $0.13 \times f_{ICLK} + 69$  (unit : mA, fCPUCLK and fICLK are MHz)

Note 3. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.

Note 4. ICLK, FCLK, PCLKA, PCLKB, PCLKC, PCLKD and PCLKE are set to divided by 64.

**Table 52.16 Increase during BGO operation (DCDC mode and External VDD mode)**

Parameter		Symbol	Typ	Max	Unit	Test conditions
Supply current <sup>*1</sup>	Data flash P/E	$I_{CC}$	6	—	mA	—
	Code flash P/E		8	—		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

**Table 52.17 Current of Low-speed mode (DCDC mode)**

Parameter	Symbol	Typ	Max	Unit	Test conditions
Supply current <sup>*1*2*3</sup>	$I_{DD}$	14.5	—	mA	CPUCLK = ICLK = 1MHz

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.

Note 3. FCLK, PCLKA, PCLKB, PCLKC, PCLKD and PCLKE are set to divided by 64 (15.6 kHz).

**Table 52.18 Standby current (DCDC mode)**

Parameter	Symbol	Typ	Max	Unit	Test conditions		
Supply current*1	Software Standby mode*2		I <sub>CC</sub>	0.02	0.94	mA	—
	Data of SRAM and TCM is retained		I <sub>CC_DCDC</sub>	0.88	28.29		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 1 (n = 0 to 6) PDRAMSCR1.RKEEP0 = 1
	Data of SRAM and TCM is not retained		I <sub>CC_DCDC</sub>	0.83	26.64		VCC_DCDC = 3.3 V PDRAMSCR0.RKEEPn = 0 (n = 0 to 6) PDRAMSCR1.RKEEP0 = 0
Deep Software Standby mode 1			I <sub>CC</sub>	5.21	148	μA	—
			I <sub>CC_DCDC</sub>	0.57	5.50		—
	Increase when the function is activated	Data of Standby SRAM is retained	I <sub>CC</sub>	0.12	2.60		—
		PVD0, PVD1,PVD2 or Battery power supply switch		See Table 52.20			—
		When the LOCO is in use		3.10	—		—
		Crystal oscillator and RTC		See Table 52.21			—
		IWDT and ULPT(all units) are operating		0.07	—		—
	Deep Software Standby mode 2		I <sub>CC</sub>	1.68	43.99		—
			I <sub>CC_DCDC</sub>	0.57	5.50		—
	Increase when the function is activated	PVD0, PVD1,PVD2 or Battery power supply switch	I <sub>CC</sub>	See Table 52.20			—
Crystal oscillator and RTC		See Table 52.21		—			
Deep Software Standby mode 3		I <sub>CC</sub>	0.99	42.90	—		
		I <sub>CC_DCDC</sub>	0.57	5.50	—		
Increase when the function is activated	Crystal oscillator and RTC		See Table 52.21		—		
					—		
RTC operating while VCC is off (with the battery backup function, only the RTC operate)	When a crystal oscillator with low power mode 3 is in use	I <sub>VBAT</sub>	0.52	—	VBATT=1.8 V, VCC=0 V		
			1.05	—	VBATT=3.3 V, VCC=0 V		
	When a crystal oscillator with low power mode 2 is in use		0.56	—	VBATT=1.8 V, VCC=0 V		
			1.10	—	VBATT=3.3 V, VCC=0 V		
	When a crystal oscillator with low power mode 1 is in use		0.62	—	VBATT=1.8 V, VCC=0V		
			1.17	—	VBATT=3.3 V, VCC=0 V		
	When a crystal oscillator with standard mode is in use		0.93	—	VBATT=1.8 V, VCC=0 V		
			1.50	—	VBATT=3.3 V, VCC=0 V		
	When EXCIN is in use		0.37	—	VBATT=1.8 V, VCC=0 V		
			0.86	—	VBATT=3.3 V, VCC=0 V		
			Increase when the function is activated	Common circuit when using RTCICn (n = 0~2) input or EXCIN	0.04	—	VBATT=1.8 V, VCC=0 V
				0.04	—	VBATT=3.3 V, VCC=0 V	
RTCICn (n = 0~2) input is in use per channel	0.02	—	VBATT=1.8 V, VCC=0 V				
	0.02	—	VBATT=3.3 V, VCC=0 V				

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Note 2. When an external clock is used, EXTAL pin is pull-up or pull-down. In case clock is toggled, software standby mode current consumption is increased by 130μA at 48MHz under typical conditions.



**Table 52.19 Coremark and normal mode current (DCDC and External power supply mode)**

Parameter			Symbol	Typ	Max	Unit	Test conditions
Supply current <sup>*1*2</sup>	Coremark	Cache off, executing from flash	I <sub>DD</sub>	165	—	μA/MHz	CPUCLK = 360 MHz ICLK = 120 MHz PCLKA = 30 MHz PCLKB = 30 MHz PCLKC = 30 MHz PCLKD = 30 MHz PCLKE = 30 MHz FCLK = 30 MHz
	Normal mode	All peripheral disabled, Cache off, while (1) code executing from flash		137	—		

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Note 2. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.

**Table 52.20 Increase when the PVD0, PVD1, PVD2 or Battery power supply switch is enabled in Deep Software Standby mode 1 and 2.**

Parameter		Symbol	Typ	Max	Unit	Test conditions
Supply current	Common circuit when enabling PVDn (n = 0 to 2) or Battery power supply switch in Deep Software Standby mode 1	I <sub>CC</sub>	4.51	—	μA	—
	Common circuit when enabling PVDn (n = 0 to 2) or Battery power supply switch in Deep Software Standby mode 2		4.97	—		—
	PVD0 enabled with OFS1(_SEC).PVDLPSEL = 1		2.16	—		—
	PVD1 enabled		2.16	—		—
	PVD2 enabled		2.16	—		—
	Battery power supply switch enabled with following conditions.*1		2.16	—		—

- Battery power supply switch enable (VBTBPCR1.BPWSWSTP = 0), voltage monitor 0 reset enable (OFS1(\_SEC).PVDAS = 0) and low power consumption function of PVD0 disable (OFS1(\_SEC).PVDLPSEL = 1).
- Battery power supply switch enable (VBTBPCR1.BPWSWSTP=0) and voltage monitor 0 reset disable (OFS1(\_SEC).PVDAS = 1).

Note 1. Consumption current is not increased in other condition.

**Table 52.21 Increase when the sub-clock oscillator and RTC are enabled in Deep Software Standby mode 1, 2 and 3.**

Parameter			Symbol	Typ	Max	Unit	Test conditions
Supply current	When a crystal oscillator is in use	Low Power mode 3	I <sub>CC</sub>	0.22	—	μA	—
		Low Power mode 2		0.27	—		—
		Low Power mode 1		0.34	—		—
		Standard mode		0.67	—		—
	RTC is operating		0.33	—	—		

**Table 52.22 Inrush current**

Parameter				Symbol	Typ	Max	Unit	Test conditions
Supply current	Inrush current on returning from deep software standby mode	Inrush current of VCC_DCD C*1	DPSBYCR.DCSSMODE = 0	I <sub>RUSH</sub>	—	630	mA	—
			DPSBYCR.DCSSMODE = 1		—	1020		—

Note 1. Reference value

**Table 52.23 Operating current (Analog)**

Parameter			Symbol	Typ	Max	Unit	Test conditions	
Supply current *1	Oscillator	Main clock oscillator	I <sub>CC</sub>	0.48	—	mA	MOMCR.MODRV0[2:0] = 000b	
				0.58	—	mA	MOMCR.MODRV0[2:0] = 011b	
				0.90	—	mA	MOMCR.MODRV0[2:0] = 101b	
	Analog power supply current	During 12-bit A/D conversion		AI <sub>CC</sub>	0.8	1.1	mA	—
		During 12-bit A/D conversion with S/H amp			2.3	3.3	mA	—
		ACMPHS(1unit)			100	150	μA	—
		Temperature sensor			0.1	0.2	mA	—
		During D/A conversion	Without AMP output		0.1	0.2	mA	—
			With AMP output		0.8	1.6	mA	—
		Waiting for A/D, D/A conversion (all units)			0.9	1.6	mA	—
		ADC12, DAC12 in standby modes (all units)*2			2	8	μA	—
	Reference power supply current (VREFH0)	During 12-bit A/D conversion (unit 0)		AI <sub>REFH0</sub>	70	120	μA	—
		Waiting for 12-bit A/D conversion (unit 0)			0.07	0.5	μA	—
		ADC12 in standby modes (unit 0)			0.07	0.5	μA	—
	Reference power supply current (VREFH)	During 12-bit A/D conversion (unit 1)		AI <sub>REFH</sub>	70	120	μA	—
		During D/A conversion	Without AMP output		0.1	0.4	mA	—
			With AMP output		0.1	0.4	mA	—
		Waiting for 12-bit A/D (unit 1), D/A (all units) conversion			0.07	0.8	μA	—
		ADC12 in standby modes (unit 1)			0.07	0.8	μA	—
	USB operating current	Low speed	USBFS	I <sub>CCUSBLS</sub>	3.5	6.5	μA	VCC_USB
		Full speed	USBFS	I <sub>CCUSBFS</sub>	4.00	10.00	mA	VCC_USB

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Note 2. When the MCU is in Software Standby mode or the MSTPCRD.MSTPD16 (12-Bit A/D Converter 0 Module Stop bit) and MSTPCRD.MSTPD15 (12-bit A/D converter 1 module stop bit) are in the module-stop state.

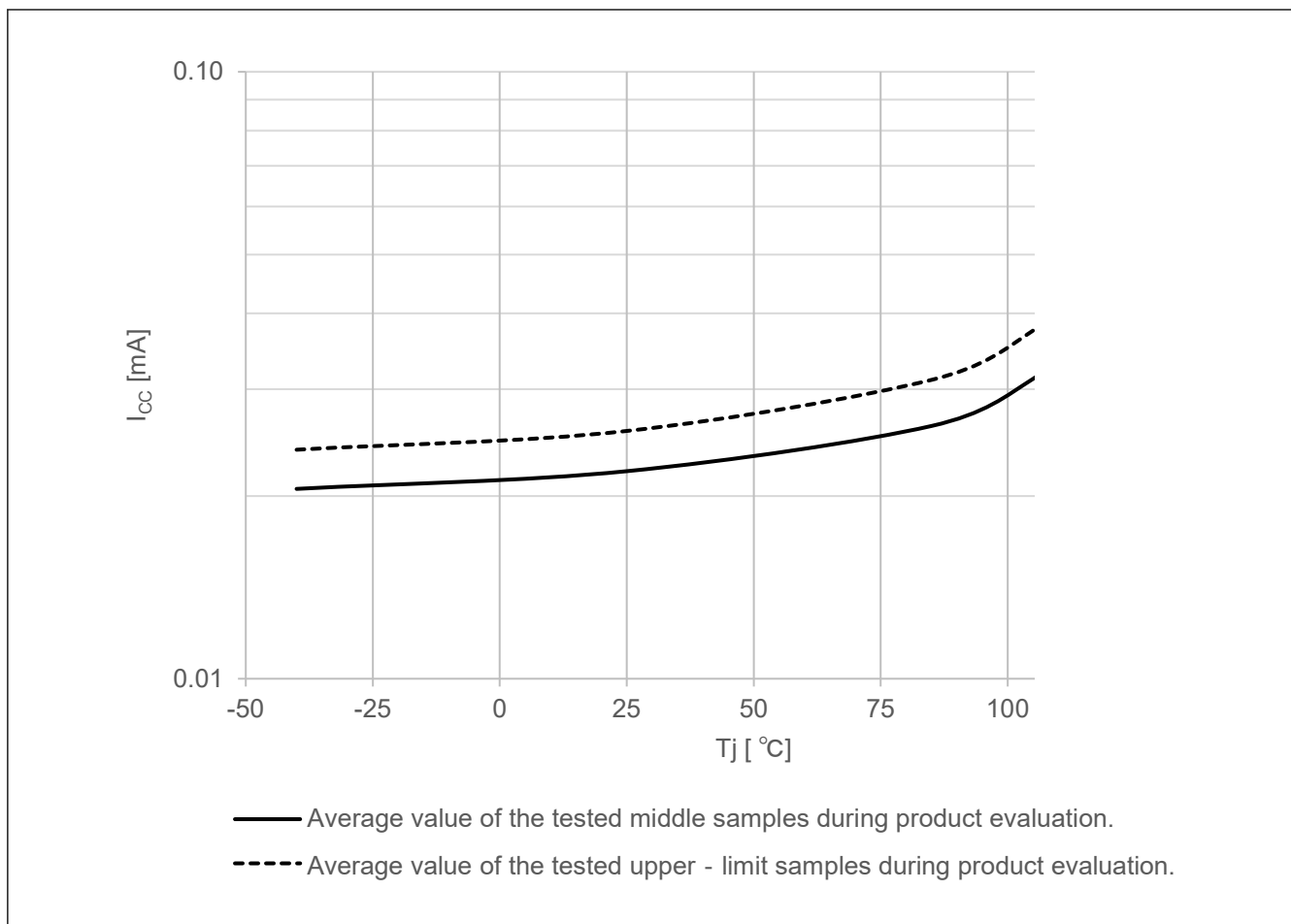


Figure 52.3 Temperature dependency in Software Standby mode ( $I_{CC}$ ) (reference data)

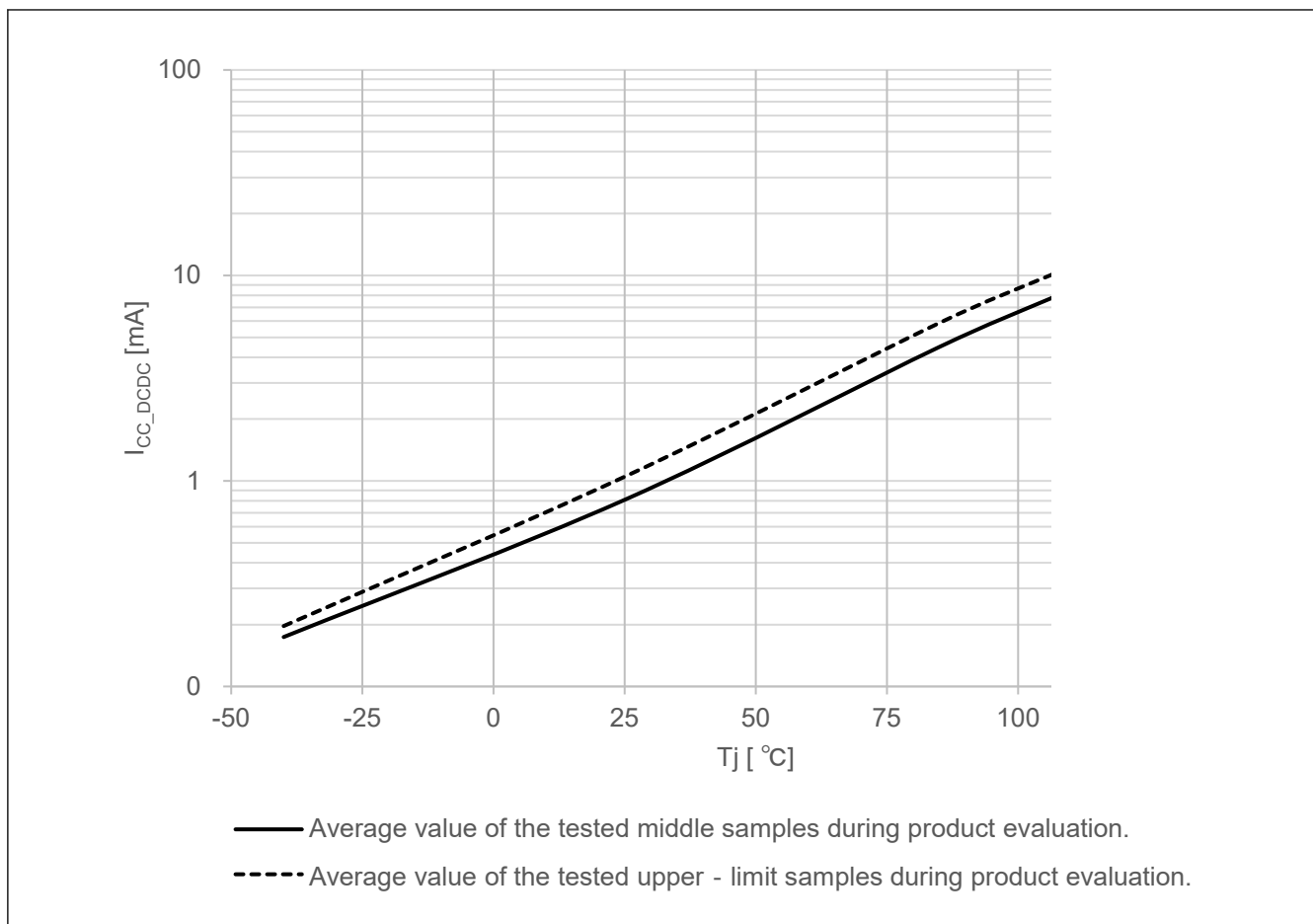


Figure 52.4 Temperature dependency in Software Standby mode ( $I_{cc\_DcDc}$ ) (reference data)

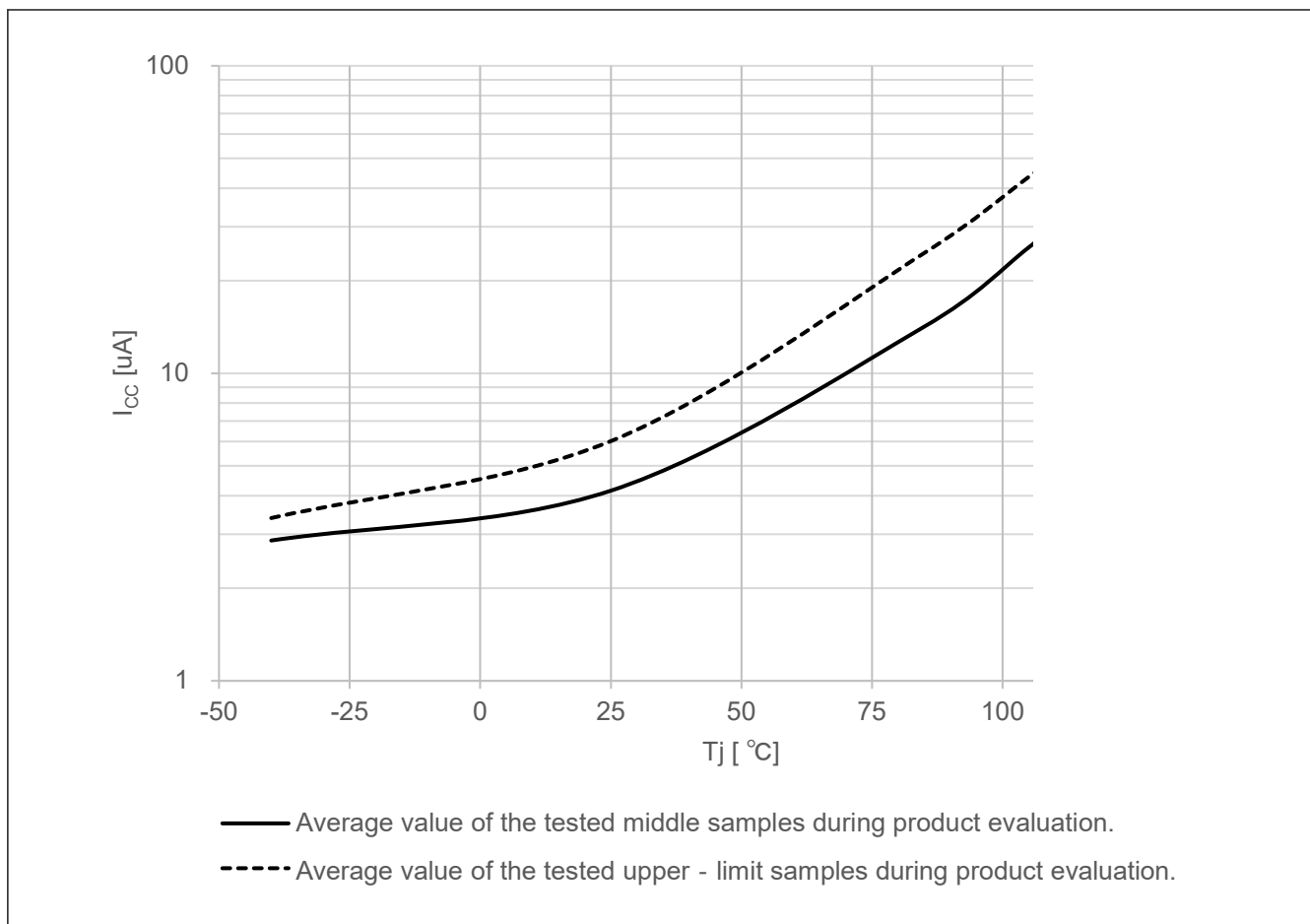


Figure 52.5 Temperature dependency in Deep Software Standby mode 1 (reference data)

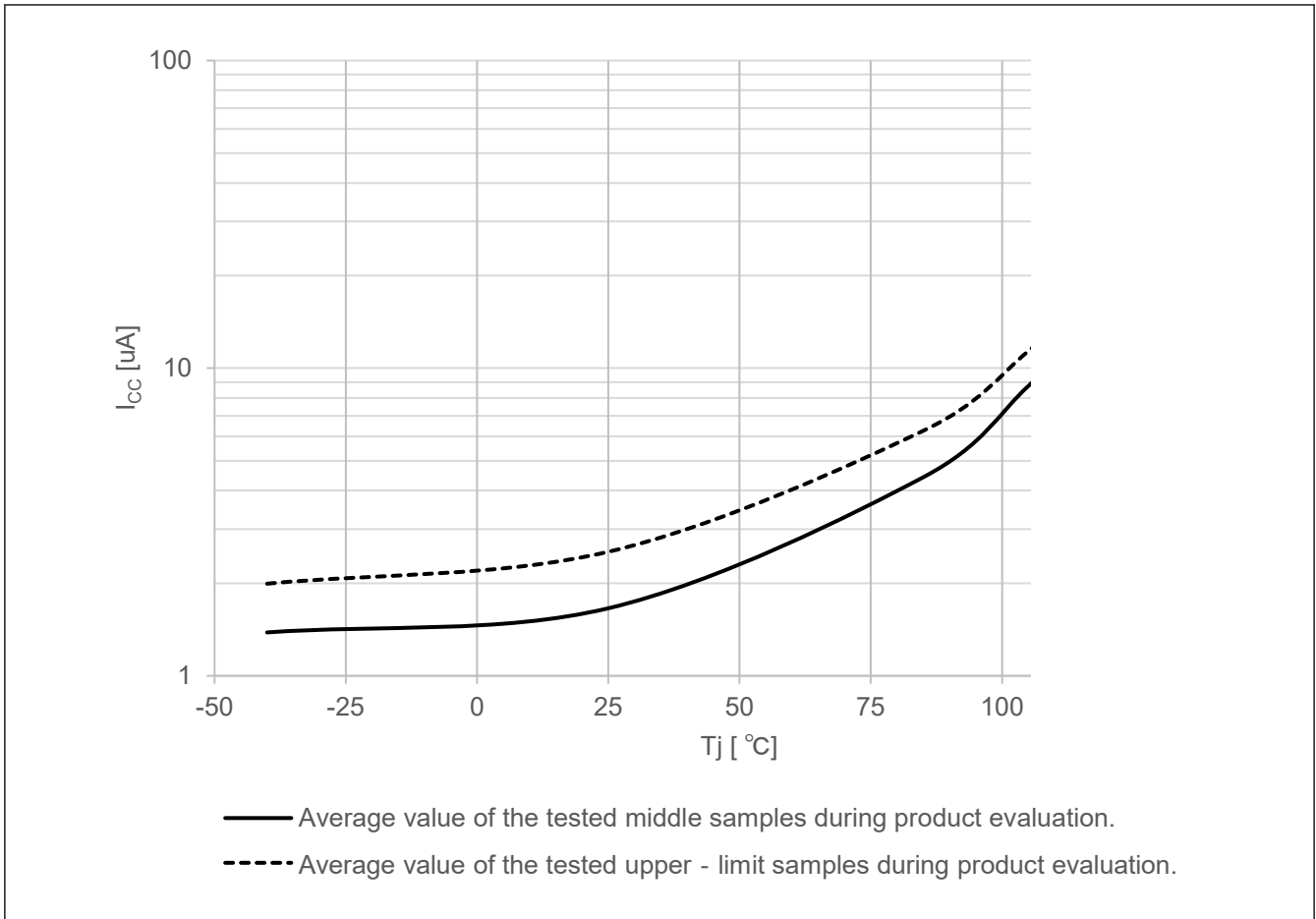


Figure 52.6 Temperature dependency in Deep Software Standby mode 2 (reference data)

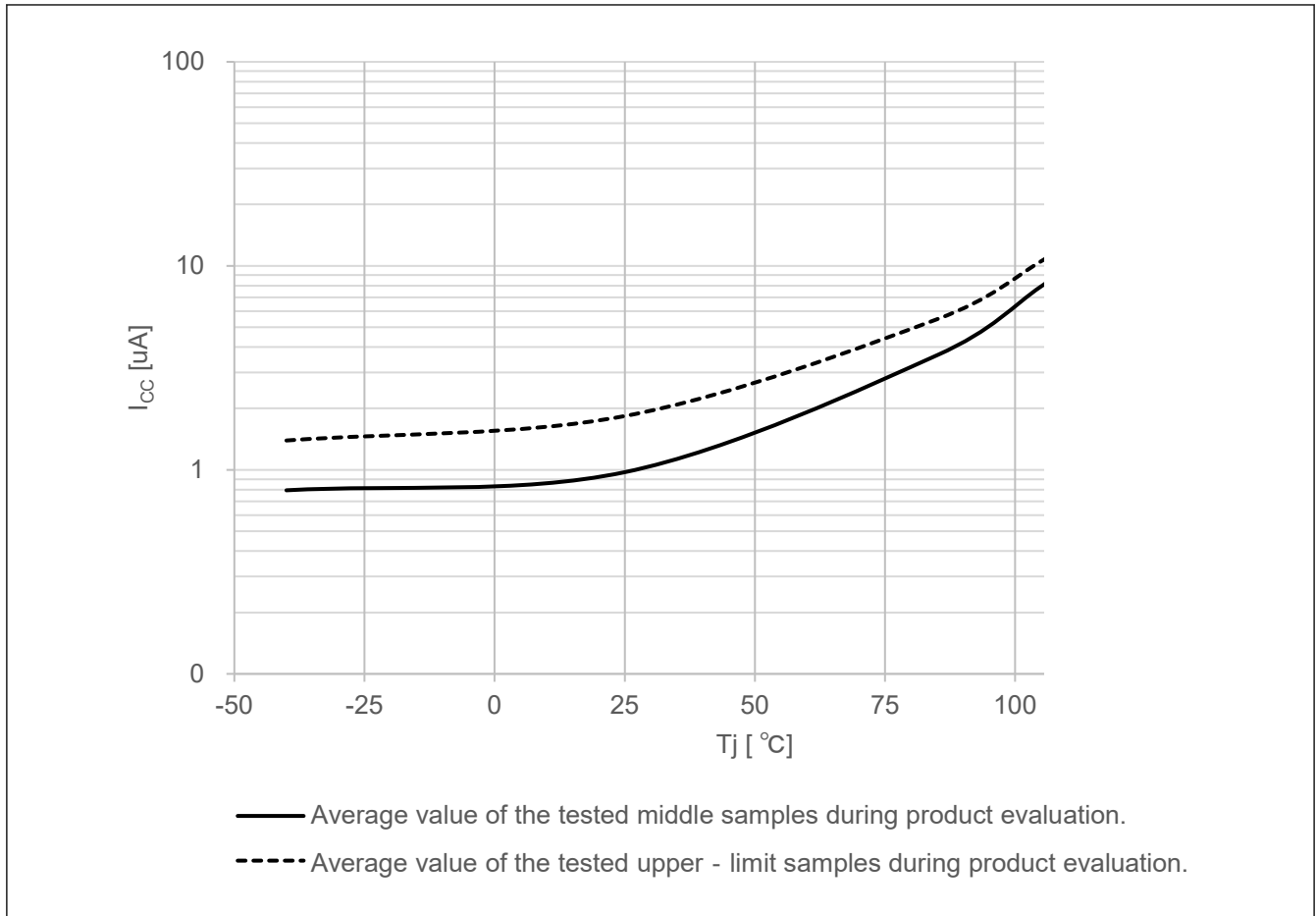


Figure 52.7 Temperature dependency in Deep Software Standby mode 3 (reference data)

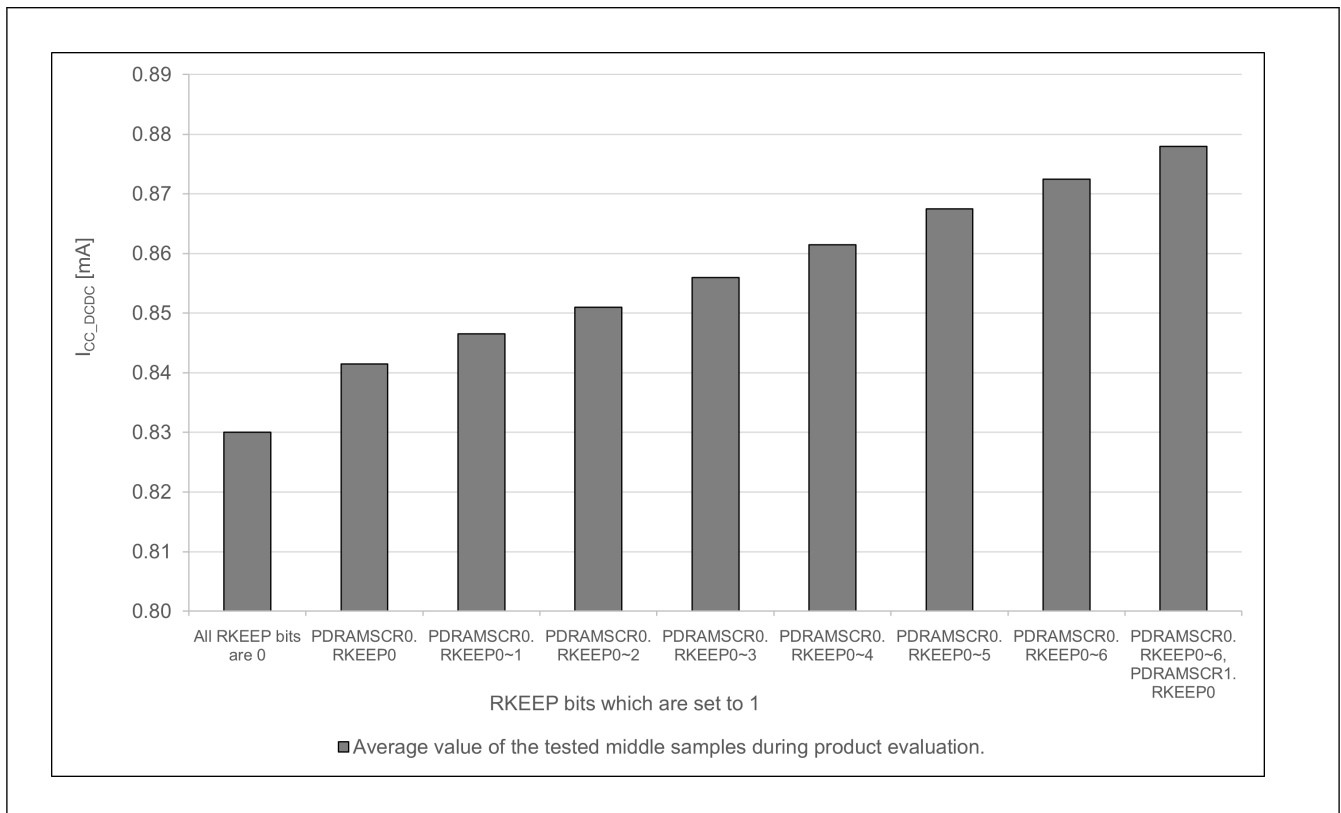


Figure 52.8 Software Standby current per SRAM state (reference data)

The more practical  $I_{CC\_DCDC}$  value can be obtained with the following formula.

$$I_{CC\_DCDC} = I_{DD} \times (V_{CL} \div V_{CC}) \div \text{efficiency}$$

Where:  $V_{CL}$  and  $V_{CC}$  are the voltage of  $V_{CL}$  pin and  $V_{CC}$  pin respectively, and efficiency is shown in the following figures.

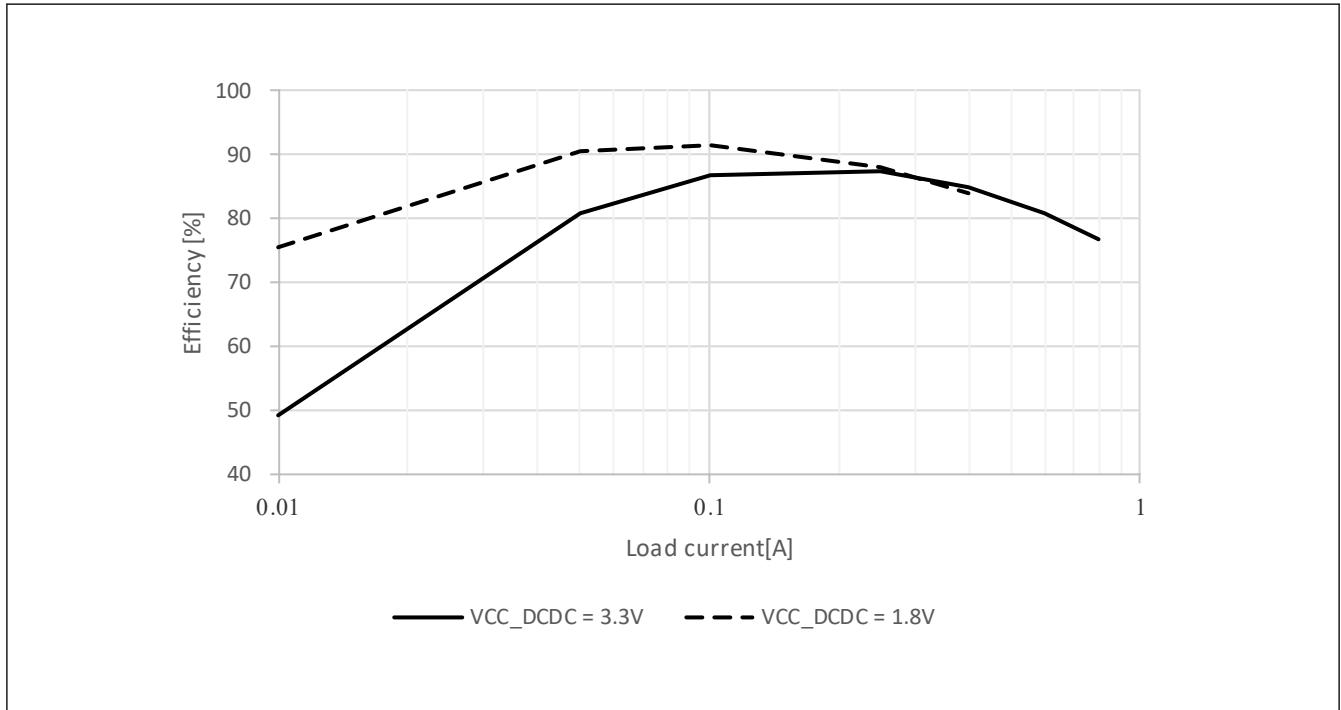


Figure 52.9 Typical DCDC efficiency (%) vs load current (A) in High-speed mode , Tj = 25°C

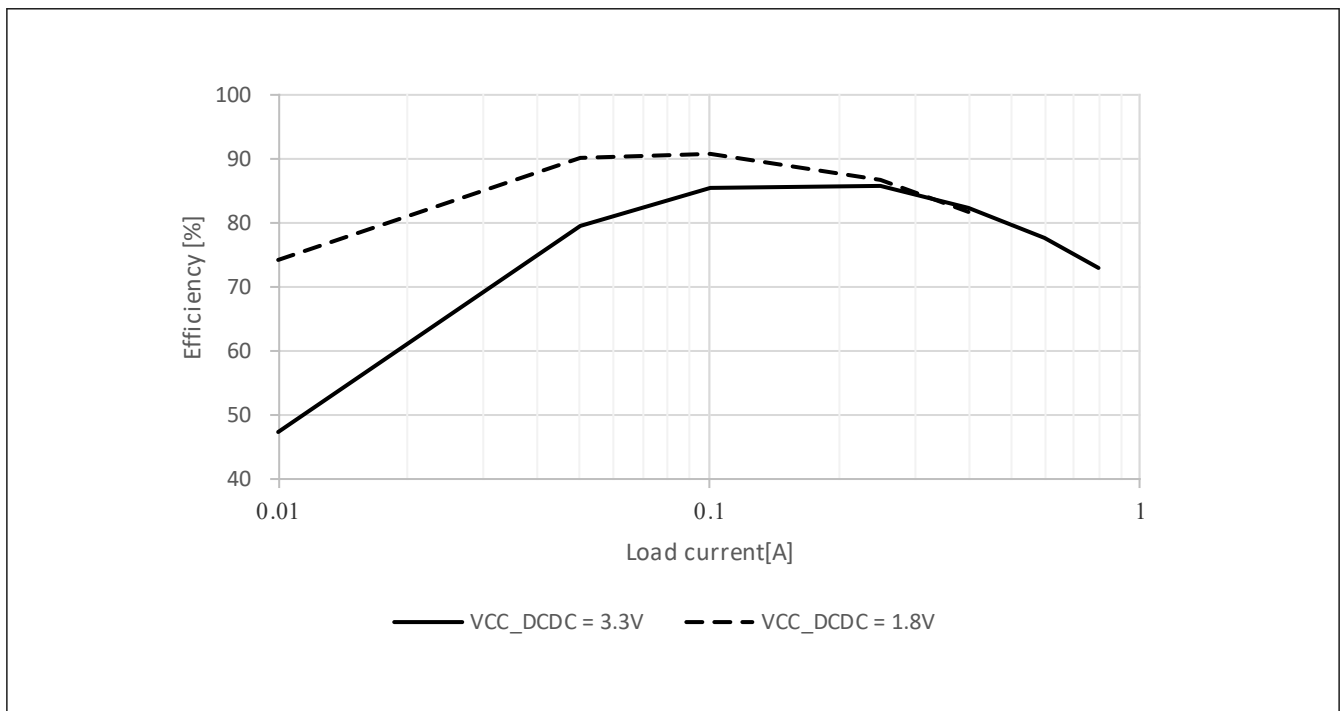
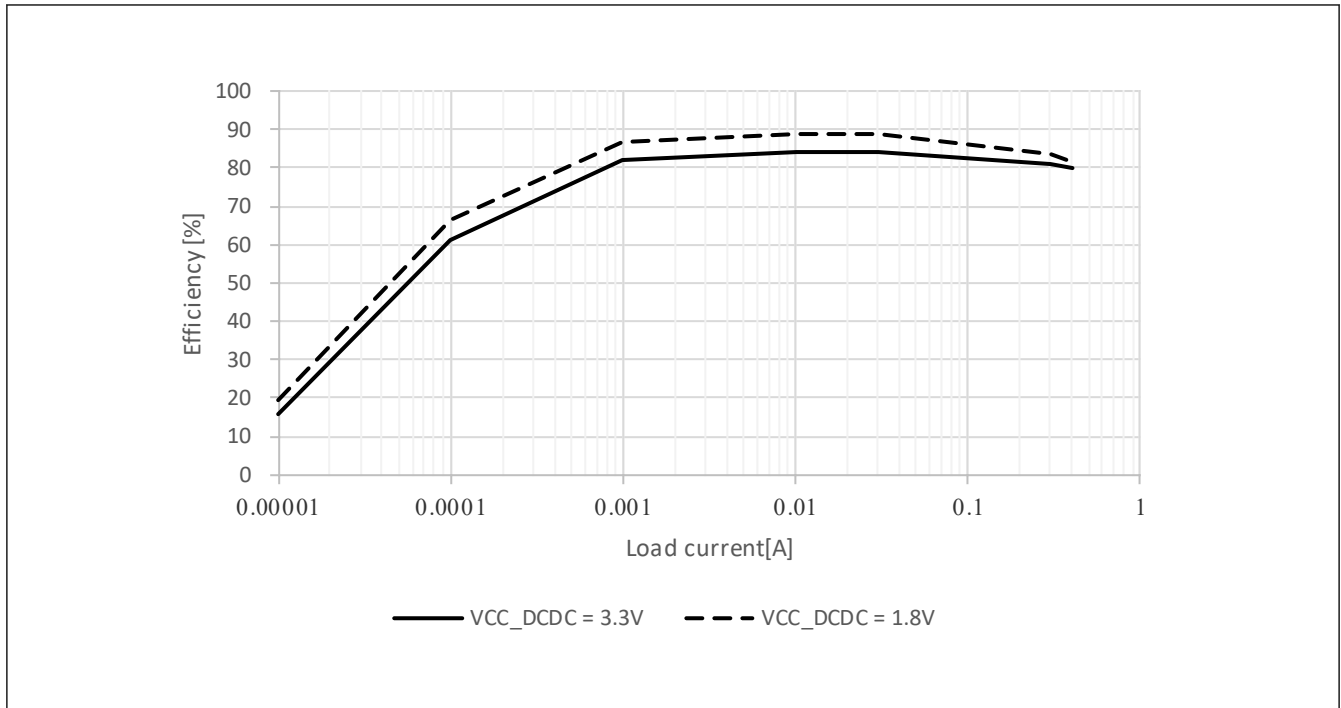
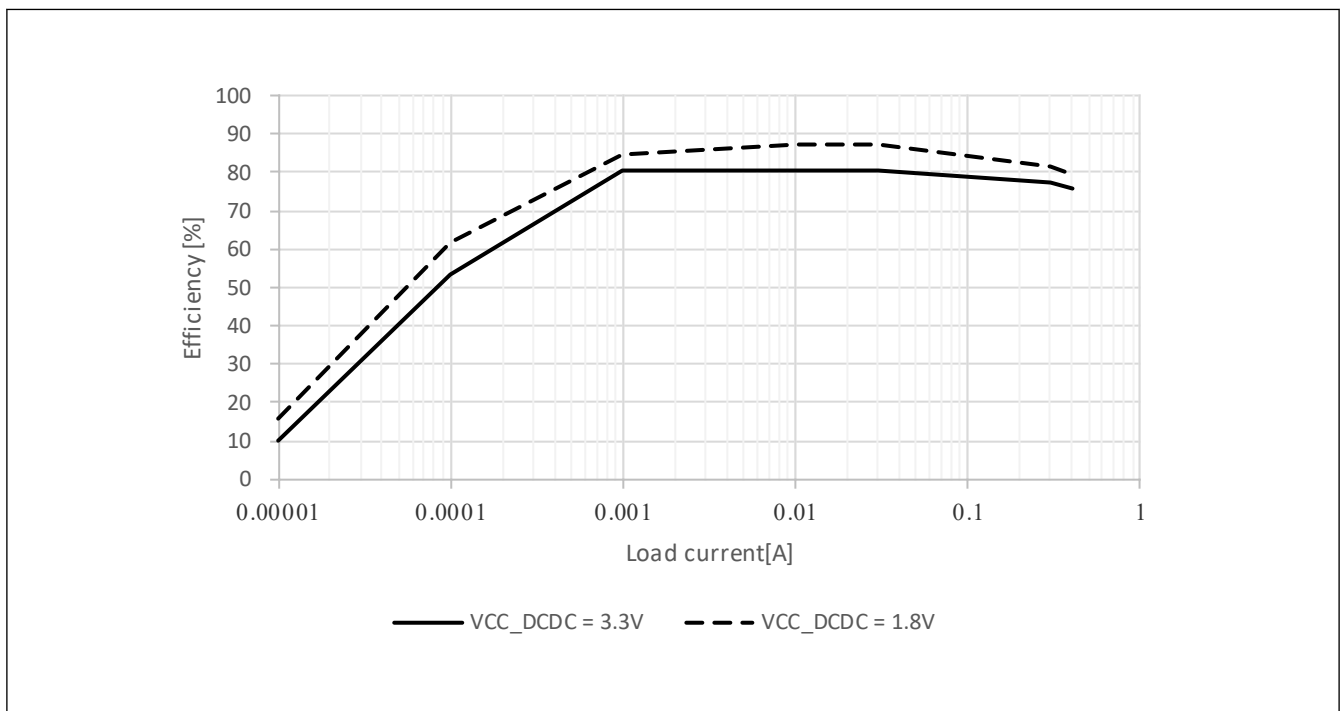


Figure 52.10 Typical DCDC efficiency (%) vs load current (A) in High-speed mode , Tj = 105°C





**Figure 52.11** Typical DCDC efficiency (%) vs load current (A) in Low-speed mode and Software Standby mode, Tj = 25°C



**Figure 52.12** Typical DCDC efficiency (%) vs load current (A) in Low-speed mode and Software Standby mode, Tj = 105°C

Note: DCDC efficiency is obtained based on the VCC\_DCDC current.

### 52.2.6 VCC Rise and Fall Gradient and Ripple Frequency

**Table 52.24 VCC rise and fall gradient characteristics at power on/off**

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions	
VCC rising gradient at power on <sup>*1</sup>	SrVCC	0.0084	—	20	ms/V	—	
VCC falling gradient at power off	VBATT function is disabled <sup>*1</sup>	SfVCC1	0.0084	—	—	ms/V	—
	VBATT function is enabled	SfVCC2	1.0000	—	—	—	—

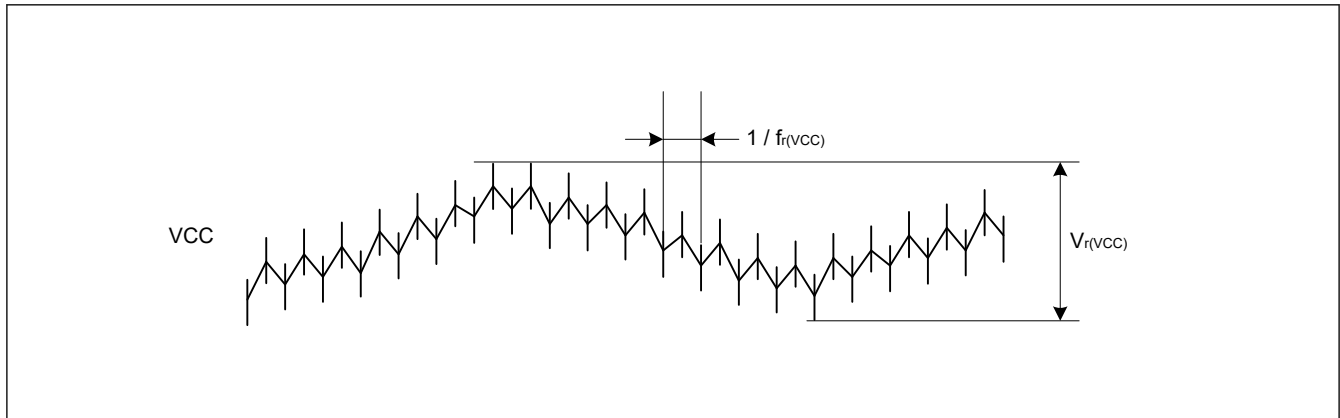
Note 1. In case the VCC voltage crosses V<sub>POR1</sub>.

**Table 52.25 VCC ripple frequency and gradient characteristics during operation**

The ripple voltage must meet the allowable ripple frequency  $f_{r(VCC)}$  within the range between the VCC upper limit (3.6 V) and lower limit (1.68 V). When the VCC change exceeds  $VCC \pm 10\%$ , the allowable voltage change rising and falling gradient  $dt/dVCC$  must be met.

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Allowable ripple frequency	$f_{r(VCC)}$	—	—	10.0	kHz	Figure 52.13 $V_{r(VCC)} \leq VCC \times 0.2$
		—	—	1.0	MHz	Figure 52.13 $V_{r(VCC)} \leq VCC \times 0.08$
		—	—	10.0	MHz	Figure 52.13 $V_{r(VCC)} \leq VCC \times 0.06$
Allowable voltage change rising and falling gradient	$dt/dVCC^{*1}$	1.0	—	—	ms/V	When VCC change exceeds $VCC \pm 10\%$

Note 1. In case the VCC voltage does not cross V<sub>POR1</sub>.



**Figure 52.13 Ripple waveform**

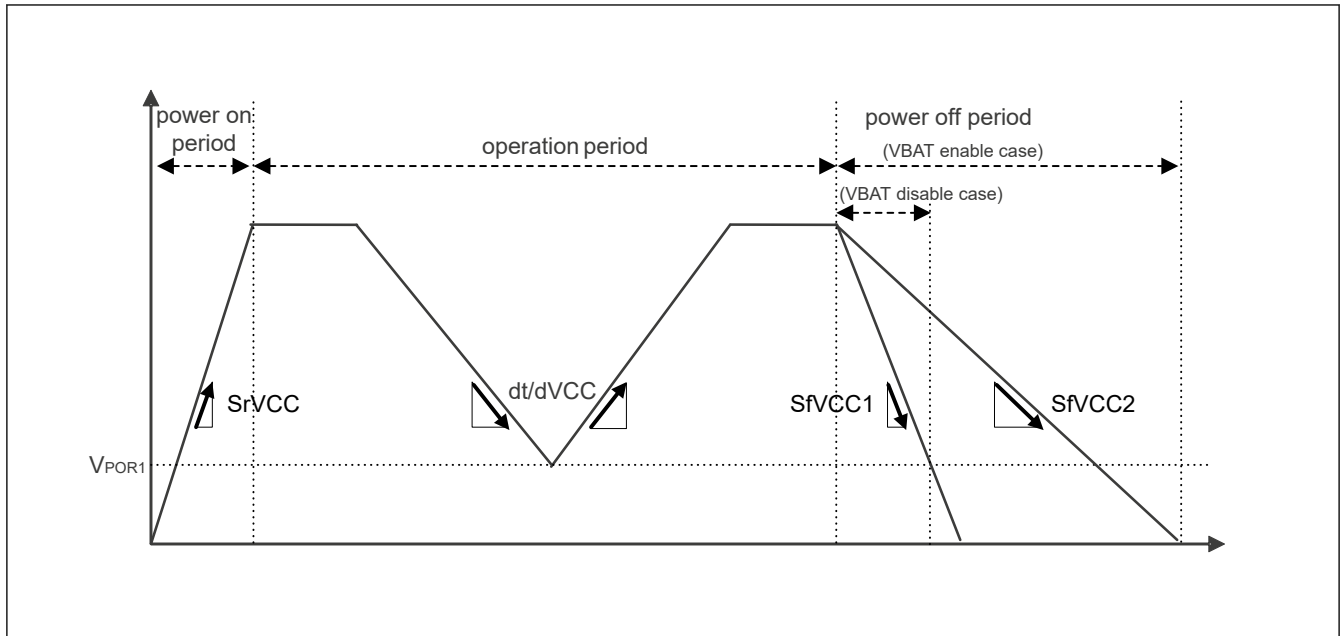


Figure 52.14 VCC rising and falling waveform

### 52.2.7 Thermal Characteristics

Maximum value of junction temperature ( $T_j$ ) must not exceed the value of [section 52.2.1.  \$T\_j/T\_a\$  Definition.](#)

$T_j$  is calculated by either of the following equations.

- $T_j = T_a + \theta_{ja} \times \text{Total power consumption}$
- $T_j = T_t + \Psi_{jt} \times \text{Total power consumption}$ 
  - $T_j$  : Junction Temperature ( $^{\circ}\text{C}$ )
  - $T_a$  : Ambient Temperature ( $^{\circ}\text{C}$ )
  - $T_t$  : Top Center Case Temperature ( $^{\circ}\text{C}$ )
  - $\theta_{ja}$  : Thermal Resistance of “Junction”-to-“Ambient” ( $^{\circ}\text{C}/\text{W}$ )
  - $\Psi_{jt}$  : Thermal Resistance of “Junction”-to-“Top Center Case” ( $^{\circ}\text{C}/\text{W}$ )
- Total power consumption = Voltage  $\times$  (Leakage current + Dynamic current)
- Leakage current of IO =  $\Sigma (I_{OL} \times V_{OL}) / \text{Voltage} + \Sigma (|I_{OH}| \times |V_{CC} - V_{OH}|) / \text{Voltage}$
- Dynamic current of IO =  $\Sigma IO (C_{in} + C_{load}) \times \text{IO switching frequency} \times \text{Voltage}$ 
  - $C_{in}$ : Input capacitance
  - $C_{load}$ : Output capacitance

Regarding  $\theta_{ja}$  and  $\Psi_{jt}$ , see [Table 52.26](#).

Table 52.26 Thermal Resistance

Parameter	Package	Symbol	Value*1	Unit	Test conditions
Thermal Resistance	100-pin LQFP (PLQP0100KP-A)	$\theta_{ja}$	32.9	$^{\circ}\text{C}/\text{W}$	JESD 51-2 and 51-7 compliant
	144-pin LQFP (PLQP0144KA-B)		31.7		
Thermal Resistance	100-pin LQFP (PLQP0100KP-A)	$\Psi_{jt}$	0.42	$^{\circ}\text{C}/\text{W}$	JESD 51-2 and 51-7 compliant
	144-pin LQFP (PLQP0144KA-B)		0.40		

Note 1. The values are reference values when the 4-layer board is used. Thermal resistance depends on the number of layers or size of the board. For details, see the JEDEC standards.

## 52.2.7.1 Calculation guide of maximum current

Table 52.27 Power consumption of each unit (1 of 2)

Dynamic current/ Leakage current	MCU Domain	Category	Item	Symbol	Frequency [MHz]	Current [ $\mu$ A/MHz]	Current*1 [mA]	Condition	
Leakage current	Analog	Regulator and Leak*1	Tj = 95°C	I <sub>CC</sub>	—	—	0.54	—	
			Tj = 105°C		—	—	0.64		
			Tj = 95°C	I <sub>CC_DCDC</sub>	—	—	62	VCC_DCDC = 3.3V, High speed mode	
			Tj = 105°C		—	—	75		
			Tj = 95°C		—	—	112		VCC_DCDC = 1.8V, High speed mode
			Tj = 105°C		—	—	134		
			Tj = 95°C	I <sub>DD</sub>	—	—	146	—	
			Tj = 105°C		—	—	175		
Dynamic current	CPU	Operation with Cache	CoreMark	I <sub>DD</sub>	360	307	111	CPUCLK = 360MHz	
	Peripheral Unit	Timer	RTC		60	1.299	0.078	—	
			GPT16 (4ch)*2		120	11.325	1.359		
			GPT32 (6ch)*2		120	15.209	1.825		
			POEG (4 Groups)*2		60	1.363	0.082		
			AGT (2ch)*2		60	2.233	0.134		
			ULPT (2ch)*2		60	0.350	0.021		
			WDT		60	0.775	0.047		
			IWDT		60	0.100	0.006		

**Table 52.27 Power consumption of each unit (2 of 2)**

Dynamic current/ Leakage current	MCU Domain	Category	Item	Symbol	Frequency [MHz]	Current [uA/MHz]	Current*1 [mA]	Condition			
Dynamic current	Peripheral Unit	Communication interfaces	ETHERC	I <sub>DD</sub>	120	8.149	0.978	—			
			USBFS		60	8.713	0.523				
			SCI (6ch)*2		120	22.717	2.726				
			IIC (2ch)*2		60	2.867	0.172				
			CANFD (2ch)*2		120	9.050	1.086				
			SPI (2ch)*2		120	7.950	0.954				
			OSPI		60	40.796	2.448				
			SSIE (2ch)*2		60	6.818	0.409				
		Analog	ADC (2Units)*2		120	3.961	0.475	—			
			DAC12		120	0.540	0.647				
			TSN		60	0.092	0.005				
			ACMPHS (2ch)*2		60	0.083	0.005				
		Human machine interfaces	CEU		120	24.143	2.897	—			
			Event link		ELC	60	1.670	0.100	—		
		Security	RSIP-E51A		120	311.301	37.4	—			
		Data processing	CRC		120	4.372	0.525	—			
			DOC		120	0.427	0.051				
		System	CAC		60	0.738	0.044	—			
		DMA	DMAC (per 1ch)		240	9.012	2.163	—			
			DTC		240	11.175	2.682				
		FSBL operation					240	—	93.4	—	
							120	—	72.9	—	

Note 1. Regulator and Leak are Internal voltage regulator's current and MCU's leakage current. It is selected according to the temperature of T<sub>j</sub>.

Note 2. To determine the current consumption per channel or unit, divide Current [mA] by the number of channels, groups or units.

**Table 52.28 Outline of operation for each unit (1 of 2)**

Peripheral	Outline of operation
RTC	RTC is operating with LOCO.
GPT	Operating modes is set to saw-wave PWM mode.
POEG	Only clear module stop bit.
AGT	AGT is operating with PCLKB.
ULPT	ULPT is operating with LOCO.
WDT	WDT is operating with PCLKB.
IWDT	IWDT is operating with IWDTCLK.
ETHERC	Operation modes is set to full-duplex mode. ETHERC is operating using Reduced Media Independent Interface (RMII).
USBFS	Transfer types is set to bulk transfer. USBFS is operating using Full-speed transfer (12 Mbps).
SCI	SCI is transmitting data in clock synchronous mode.
IIC	Communication format is set to I2C-bus format. IIC is transmitting data in master mode.

**Table 52.28 Outline of operation for each unit (2 of 2)**

Peripheral	Outline of operation
CANFD	CANFD is transmitting and receiving data in self-test mode 1.
SPI	SPI mode is set to SPI operation (4-wire method). SPI master/slave mode is set to master mode. SPI is transmitting 32-bit width data.
OSPI	OSPI is issuing memory write command to HyperRAM.
SSIE	Communication mode is set to Master. System word length is set to 32 bits. Data word length is set to 20 bits. SSIE is transmitting data using I2S format.
ADC	Resolution is set to 12-bit accuracy. Data registers is set to A/D-converted value addition mode. ADC12 is converting the analog input in continuous scan mode.
DAC12	DAC12 is outputting the conversion result while updating the value of data register.
TSN	TSN is operating.
ACMPHS	ACMPHS is operating.
CEU	CEU is capturing data and transferring to the SRAM.
ELC	Only clear module stop bit.
RSIP-E51A	RSIP is doing self-test operation.
CRC	CRC is generating CRC code using 32-bit CRC32-C polynomial.
DOC	DOC is operating in data comparison mode.
CAC	Measurement target clocks is set to PCLKB. Measurement reference clocks is set to PCLKB. CAC is measuring the clock frequency accuracy.
DMAC	Bit length of transfer data is set to 32 bits. Transfer mode is set to block transfer mode. DMAC is transferring data from SRAM to SRAM.
DTC	Bit length of transfer data is set to 32 bits. Transfer mode is set to block transfer mode. DTC is transferring data from SRAM to SRAM.

### 52.2.7.2 Example of Tj calculation

Assumption :

- Package 144-pin LQFP :  $\theta_{ja} = 31.7 \text{ }^\circ\text{C/W}$
- $T_a = 60 \text{ }^\circ\text{C}$
- $I_{CC} + I_{CC\_DCDC} = 240 \text{ mA}$
- $V_{CC} = 3.5 \text{ V}$  ( $V_{CC} = V_{CC2} = AV_{CC0} = V_{CC\_USB}$ )
- $I_{OH} = 1 \text{ mA}$ ,  $V_{OH} = V_{CC} - 0.5 \text{ V}$ , 12 Outputs
- $I_{OL} = 20 \text{ mA}$ ,  $V_{OL} = 1.0 \text{ V}$ , 8 Outputs
- $I_{OL} = 1 \text{ mA}$ ,  $V_{OL} = 0.5 \text{ V}$ , 12 Outputs
- $C_{in} = 8 \text{ pF}$ , 32 pins, Input frequency = 10 MHz
- $C_{load} = 30 \text{ pF}$ , 32 pins, Output frequency = 10 MHz

Static current of IO =  $\Sigma (V_{OL} \times I_{OL}) / \text{Voltage} + \Sigma ((V_{CC} - V_{OH}) \times I_{OH}) / \text{Voltage}$

$$\begin{aligned}
 &= (20 \text{ mA} \times 1 \text{ V}) \times 8 / 3.5 \text{ V} + (1 \text{ mA} \times 0.5 \text{ V}) \times 12 / 3.5 \text{ V} + ((V_{CC} - (V_{CC} - 0.5 \text{ V})) \times 1 \text{ mA}) \times 12 / 3.5 \text{ V} \\
 &= 45.7 \text{ mA} + 1.71 \text{ mA} + 1.71 \text{ mA} \\
 &= 49.1 \text{ mA}
 \end{aligned}$$

Dynamic current of IO =  $\Sigma IO (C_{in} + C_{load}) \times IO \text{ switching frequency} \times \text{Voltage}$

$$\begin{aligned}
 &= ((8 \text{ pF} \times 32) \times 10 \text{ MHz} + (30 \text{ pF} \times 32) \times 10 \text{ MHz}) \times 3.5 \text{ V} \\
 &= 42.6 \text{ mA}
 \end{aligned}$$

Total power consumption = Voltage  $\times$  (Static current + Dynamic current)

$$= (240 \text{ mA} \times 3.5 \text{ V}) + (49.1 \text{ mA} + 42.6 \text{ mA}) \times 3.5 \text{ V}$$

$$= 1161 \text{ mW (1.161 W)}$$

$$T_j = T_a + \theta_{ja} \times \text{Total power consumption}$$

$$= 60 \text{ }^\circ\text{C} + 31.7 \text{ }^\circ\text{C/W} \times 1.161 \text{ W}$$

$$= 96.8 \text{ }^\circ\text{C}$$

## 52.3 AC Characteristics

### 52.3.1 Frequency

**Table 52.29 Operation frequency value in high-speed mode**

Parameter	Symbol	Min	Typ	Max	Unit
Operation frequency	CPU clock (CPUCLK) <sup>*2</sup> DCDC, 144-pin LQFP and 100-pin LQFP package, T <sub>j</sub> ≤ 105°C <sup>*4</sup>	—	—	360	MHz
		—	—	360	
	External VDD, 144-pin LQFP and 100-pin LQFP package, T <sub>j</sub> ≤ 105°C	—	—	360	
	System clock (ICLK) <sup>*2</sup>	—	—	240	
	Peripheral module clock (PCLKA) <sup>*2</sup>	—	—	120	
	Peripheral module clock (PCLKB) <sup>*2</sup>	—	—	60	
	Peripheral module clock (PCLKC) <sup>*2</sup>	— <sup>*3</sup>	—	60	
	Peripheral module clock (PCLKD) <sup>*2</sup>	—	—	120	
	Peripheral module clock (PCLKE) <sup>*2</sup>	—	—	240	
	Flash interface clock (FCLK) <sup>*2</sup>	— <sup>*1</sup>	—	60	
	SCI clock (SCICLK)	—	—	120	
	SPI clock (SPICLK)	—	—	120	
	Octal SPI clock (OCTACLK)	—	—	200	
	CANFD core clock (CANFDCLK)	—	—	80	
	USB clock (USBCLK)	—	—	48	

Note 1. FCLK must run at a frequency of at least 4 MHz when programming or erasing the flash memory.

Note 2. See [section 8, Clock Generation Circuit](#) for the relationship between the CPUCLK, ICLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE, FCLK frequencies.

Note 3. When the ADC12 is used, the PCLKC frequency must be at least 1 MHz.

Note 4. When DCDC is used with VCC\_DCDC < 2.5V, I<sub>DD</sub> current must be less than the value specified in operating current.

Please see [Table 52.8](#).

**Table 52.30 Operation frequency value in low-speed mode**

Parameter	Symbol	Min	Typ	Max	Unit	
Operation frequency	CPU clock (CPUCLK) <sup>*2</sup>	f	—	—	1	MHz
	System clock (ICLK) <sup>*2</sup>	—	—	—	1	
	Peripheral module clock (PCLKA) <sup>*2</sup>	—	—	—	1	
	Peripheral module clock (PCLKB) <sup>*2</sup>	—	—	—	1	
	Peripheral module clock (PCLKC) <sup>*2</sup>	— <sup>*3</sup>	—	—	1	
	Peripheral module clock (PCLKD) <sup>*2</sup>	—	—	—	1	
	Peripheral module clock (PCLKE) <sup>*2</sup>	—	—	—	1	
	Flash interface clock (FCLK) <sup>*2</sup>	— <sup>*1</sup>	—	—	1	
	SCI clock (SCICLK)	—	—	—	1	
	SPI clock (SPICLK)	—	—	—	1	
	Octal SPI clock (OCTACLK)	—	—	—	1	
	CANFD core clock (CANFDCLK)	—	—	—	1	
	USB clock (USBCLK)	—	—	—	1	

Note 1. Programming or erasing the flash memory is disabled in low-speed mode.

Note 2. See [section 8, Clock Generation Circuit](#) for the relationship between the CPUCLK, ICLK, PCLKA, PCLKB, PCLKC, PCLKD, PCLKE, FCLK frequencies.

Note 3. When the ADC12 is used, the PCLKC frequency must be at least 1 MHz.

## 52.3.2 Clock Timing

**Table 52.31 Clock timing except for sub-clock oscillator (1 of 2)**

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
EXTAL external clock input cycle time	t <sub>EXcyc</sub>	20.80	—	—	ns	Figure 52.15
EXTAL external clock input high pulse width	t <sub>EXH</sub>	5.30	—	—	ns	
EXTAL external clock input low pulse width	t <sub>EXL</sub>	5.30	—	—	ns	
EXTAL external clock rise time	t <sub>EXr</sub>	—	—	3.0	ns	
EXTAL external clock fall time	t <sub>EXf</sub>	—	—	3.0	ns	
Main clock oscillator frequency	f <sub>MAIN</sub>	8	—	48	MHz	—
Main clock oscillation stabilization wait time (crystal) <sup>*1</sup>	t <sub>MAINOSCWT</sub>	—	—	— <sup>*1</sup>	ms	Figure 52.16
LOCO clock oscillation frequency	f <sub>LOCO</sub>	29.4912	32.768	36.0448	kHz	—
LOCO clock oscillation stabilization wait time	t <sub>LOCOWT</sub>	—	—	60.4	μs	Figure 52.17
MOCO clock oscillation frequency	f <sub>MOCO</sub>	6.8	8.0	9.2	MHz	—
MOCO clock oscillation stabilization wait time	t <sub>MOCOWT</sub>	—	—	15.0	μs	—



**Table 52.31 Clock timing except for sub-clock oscillator (2 of 2)**

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions	
HOCO clock oscillator oscillation frequency	Without FLL	$f_{\text{HOCO16}}$	15.78	16.00	16.22	MHz	$-20 \leq T_j \leq 105 \text{ }^\circ\text{C}$	
		$f_{\text{HOCO18}}$	17.75	18.00	18.25			
		$f_{\text{HOCO20}}$	19.72	20.00	20.28			
		$f_{\text{HOCO32}}$	31.55	32.00	32.45			
		$f_{\text{HOCO48}}$	47.33	48.00	48.67			
		$f_{\text{HOCO16}}$	15.71	16.00	16.29			$-40 \leq T_j \leq -20 \text{ }^\circ\text{C}$
		$f_{\text{HOCO18}}$	17.68	18.00	18.32			
		$f_{\text{HOCO20}}$	19.64	20.00	20.36			
		$f_{\text{HOCO32}}$	31.42	32.00	32.58			
		$f_{\text{HOCO48}}$	47.14	48.00	48.86			
	With FLL	$f_{\text{HOCO16}}$	15.960	16.000	16.040	MHz	$-40 \leq T_j \leq 105 \text{ }^\circ\text{C}$ Sub-clock frequency accuracy is $\pm 50$ ppm.	
		$f_{\text{HOCO18}}$	17.955	18.000	18.045			
		$f_{\text{HOCO20}}$	19.950	20.000	20.050			
		$f_{\text{HOCO32}}$	31.920	32.000	32.080			
		$f_{\text{HOCO48}}$	47.880	48.000	48.120			
HOCO clock oscillation stabilization wait time <sup>*2</sup>		$t_{\text{HOCOWT}}$	—	—	64.7	$\mu\text{s}$	—	
HOCO stop width time		$t_{\text{HOCOSTP}}$	1	—	—	$\mu\text{s}$	Figure 52.20	
HOCO period jitter		Jitter	-3	—	3	%	—	
FLL stabilization wait time		$t_{\text{FLLWT}}$	—	—	1.8	ms	—	
PLL1/PLL2 VCO frequency		$f_{\text{VCO}}$	640	—	1440.0	MHz	—	
PLL1/PLL2 Output frequency for output clock P		$t_{\text{PLL}}$	40	—	480	MHz	—	
PLL1/PLL2 Output frequency for output clock Q, R		$t_{\text{PLL}}$	71	—	480	MHz	—	
PLL1/PLL2 clock oscillation stabilization wait time		$t_{\text{PLLWT}}$	—	—	40	$\mu\text{s}$	Figure 52.18	
PLL1/PLL2 period jitter		—	—	$\pm 70$	—	ps	—	
PLL1/PLL2 long term jitter		—	—	$\pm 300$	—	ps	Term: 1 $\mu\text{s}$ , 10 $\mu\text{s}$	

Note 1. When setting up the main clock oscillator, ask the oscillator manufacturer for an oscillation evaluation, and use the results as the recommended oscillation stabilization time. Set the MOSCWTCR register to a value equal to or greater than the recommended value.

After changing the setting in the MOSCCR.MOSTP bit to start main clock operation, read the OSCSF.MOSCSF flag to confirm that it is 1, and then start using the main clock oscillator.

Note 2. This is the time from release from reset state until the HOCO oscillation frequency ( $f_{\text{HOCO}}$ ) reaches the range for guaranteed operation.

**Table 52.32 Clock timing for the sub-clock oscillator**

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Sub-clock frequency	$f_{\text{SUB}}$	—	32.768	—	kHz	—
Sub-clock oscillation stabilization wait time		$t_{\text{SUBOSCWT}}$	—	—	$\mu\text{s}$ <sup>*1</sup>	Figure 52.19

Note 1. When setting up the sub-clock oscillator, ask the oscillator manufacturer for an oscillation evaluation and use the results as the recommended oscillation stabilization time.

After changing the setting in the SOSCCR.SOSTP bit to start sub-clock operation, only start using the sub-clock oscillator after the sub-clock oscillation stabilization time elapses with an adequate margin. A value that is two times the value shown is recommended.

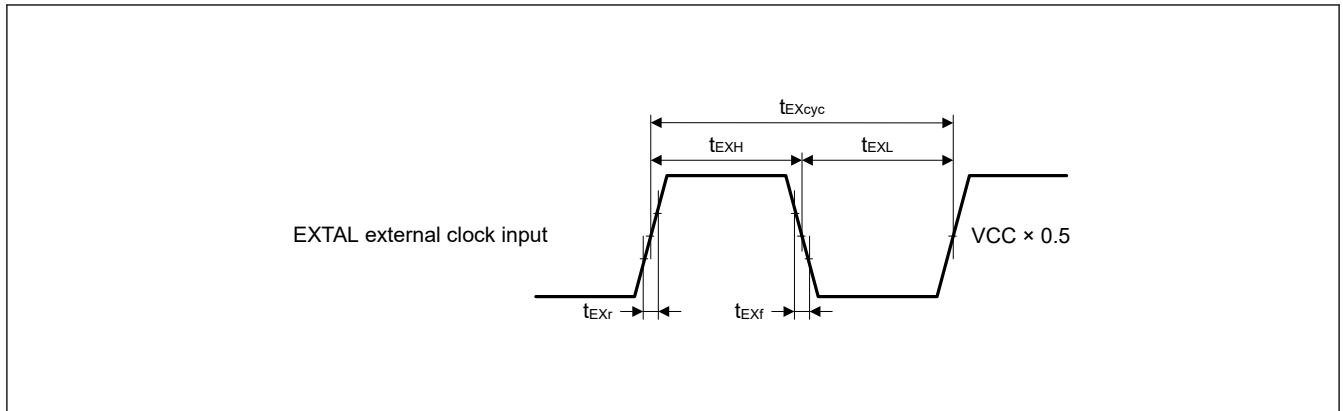


Figure 52.15 EXTAL external clock input timing

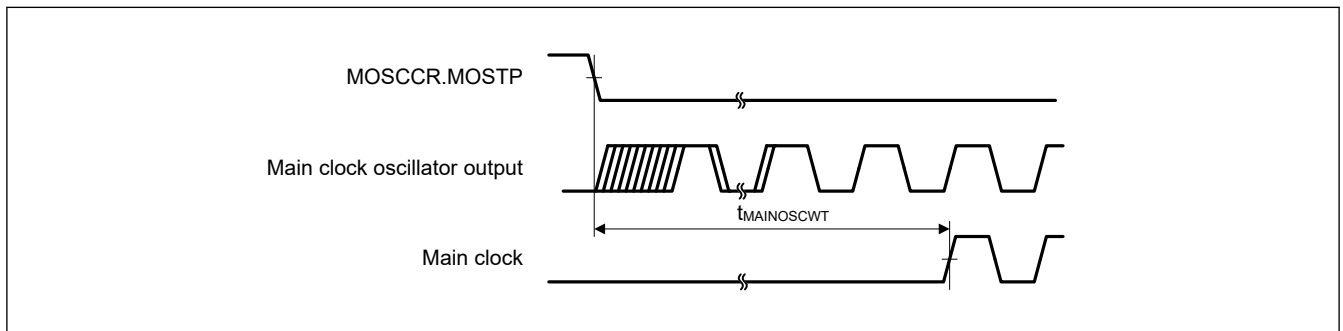


Figure 52.16 Main clock oscillation start timing

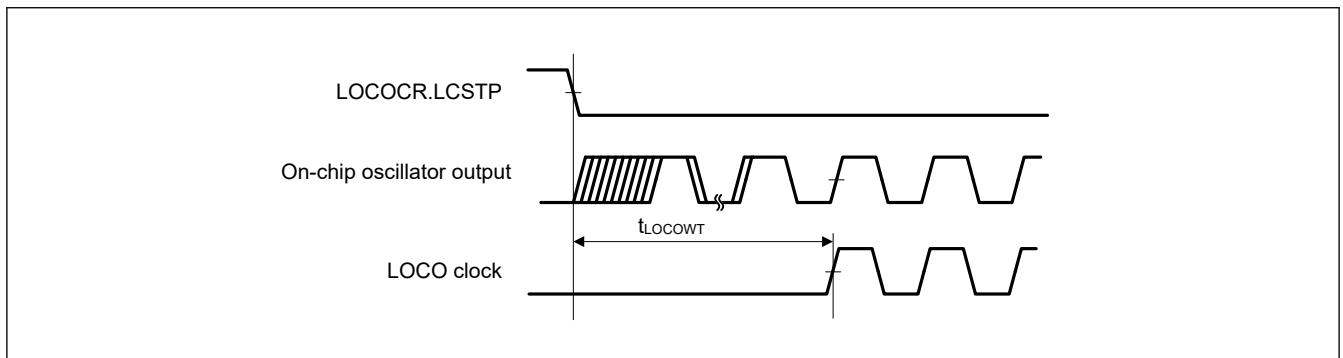


Figure 52.17 LOCO clock oscillation start timing

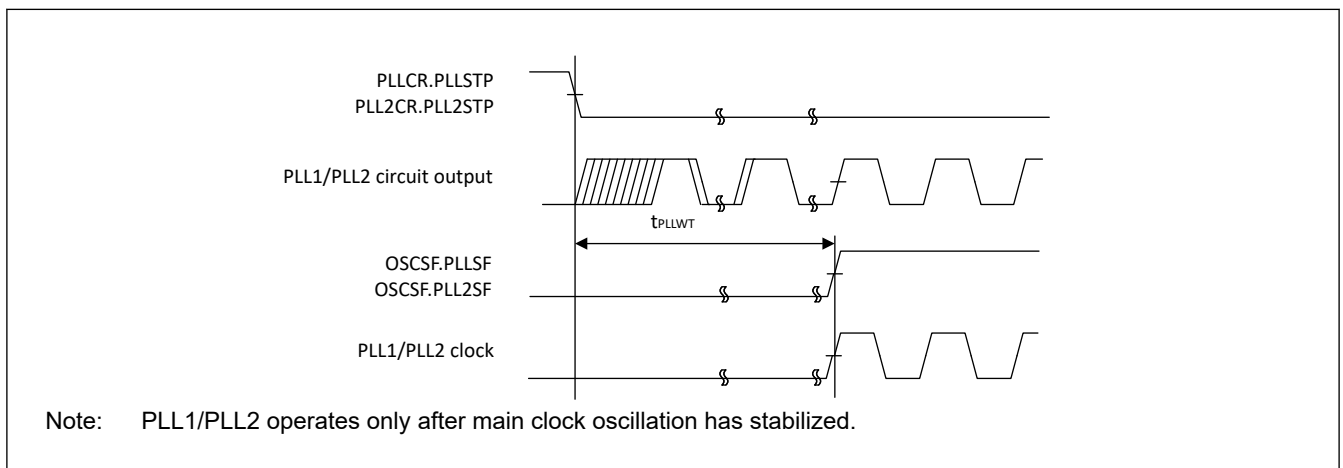


Figure 52.18 PLL1/PLL2 clock oscillation start timing

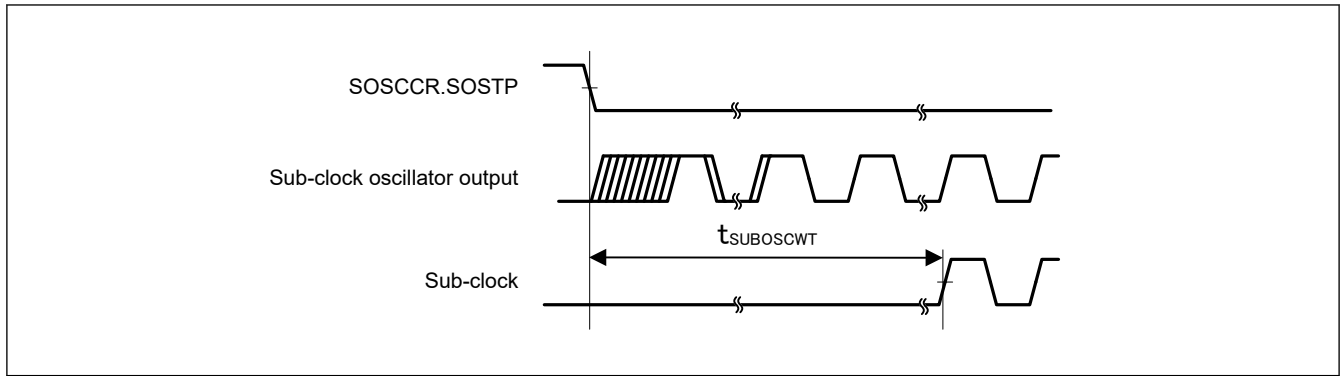


Figure 52.19 Sub-clock oscillation start timing

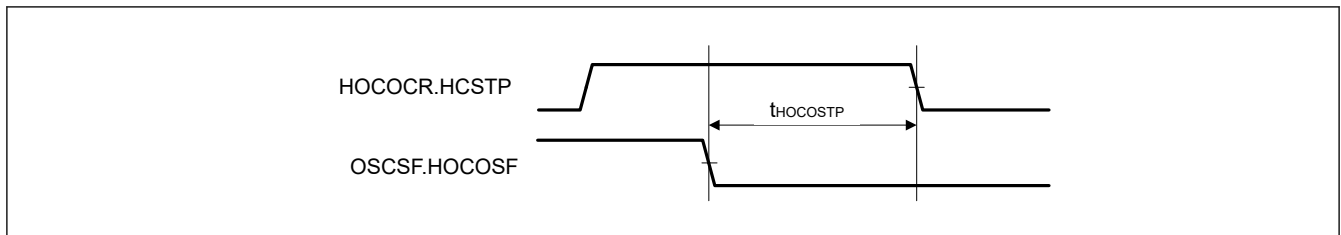


Figure 52.20 HOCO stop width time

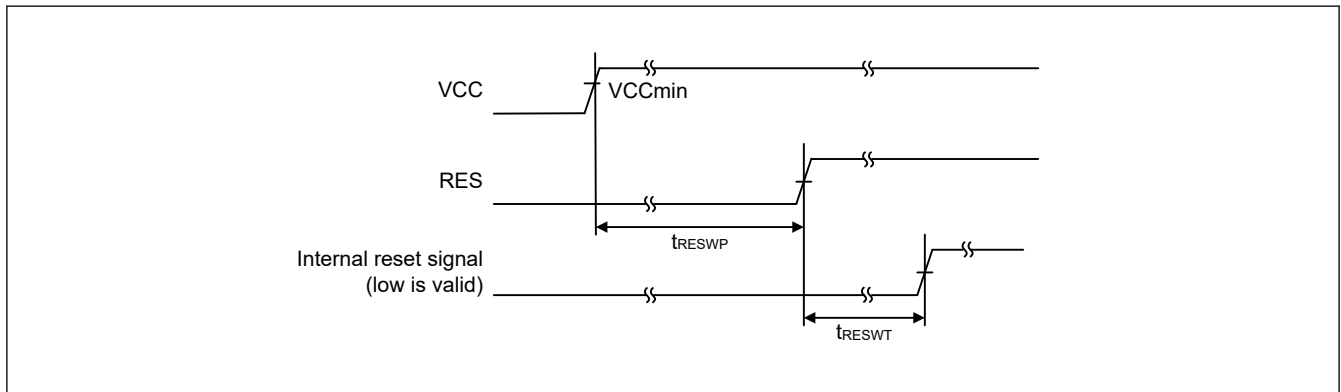
### 52.3.3 Reset Timing

Table 52.33 Reset timing (1 of 2)

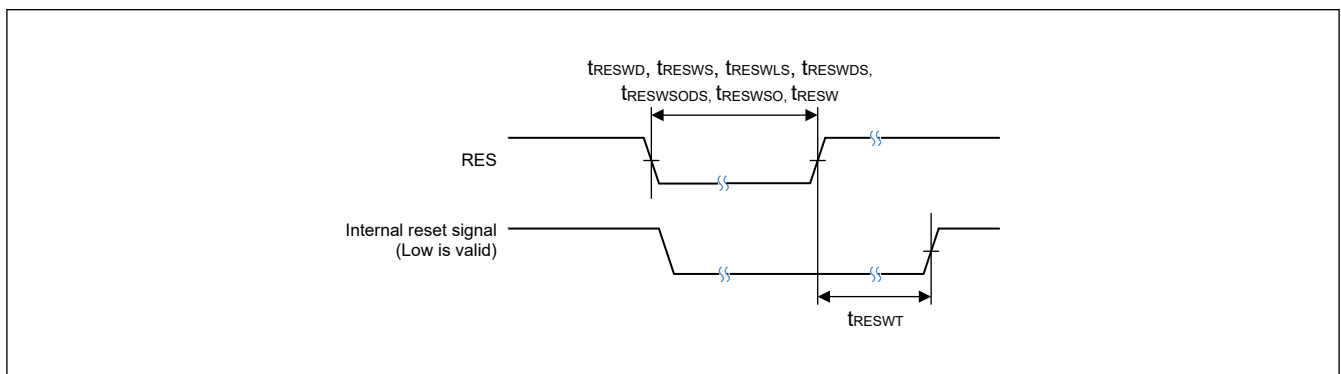
Parameter	Symbol	Min	Typ	Max	Unit	Test conditions		
RES pulse width	Power-on	$t_{RESWP}$	4.2	—	—	ms	Figure 52.21	
RES pulse width	Deep Software Standby mode 1	DPSBYCR.DCSSMODE = 0	$t_{RESWD}$	1.30	—	—	ms	Figure 52.22
		DPSBYCR.DCSSMODE = 1	0.71	—	—			
	Deep Software Standby mode 2	DPSBYCR.DCSSMODE = 0	2.00	—	—			
		DPSBYCR.DCSSMODE = 1	1.50	—	—			
	Deep Software Standby mode 3	DPSBYCR.DCSSMODE = 0	3.50	—	—			
		DPSBYCR.DCSSMODE = 1	2.90	—	—			
	Software Standby mode	$t_{RESWS}$	0.66	—	—	ms		
	Low-speed Mode	$t_{RESWLS}$	0.46	—	—	ms		
	CPU Deep Sleep mode (SOSC operation)	$t_{RESWSODS}$	0.36	—	—	ms		
	CPU Deep Sleep mode (Other than SOSC operation)	$t_{RESWDS}$	0.24	—	—	ms		
SOSC operation	$t_{RESWSO}$	0.19	—	—	ms			
Other than above	$t_{RESW}$	62.0	—	—	$\mu$ s			
Wait time after RES cancellation	$t_{RESWT}$	—	54.9	64.6	$\mu$ s	Figure 52.21		

**Table 52.33 Reset timing (2 of 2)**

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Wait time after internal reset cancellation (IWDT reset, WDT reset, CPU Lockup reset, Bus Error reset, Common Memory Error reset, Software reset)	$t_{RESW2}$	—	54.9	64.6	$\mu\text{s}$	—



**Figure 52.21 RES pin input timing under the condition that VCC exceeds  $V_{PDR}$  voltage threshold**



**Figure 52.22 Reset input timing**

### 52.3.4 Wakeup Timing

**Table 52.34 Timing of recovery from low power modes (1 of 2)**

Parameter	Fast return function <sup>*9</sup>	Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from CPU Deep Sleep mode	—	$t_{DSLPL}^{*11}$	—	182	214	$\mu\text{s}$	—

**Table 52.34 Timing of recovery from low power modes (2 of 2)**

Parameter			Fast return function <sup>*9</sup>	Symbol	Min	Typ	Max	Unit	Test conditions
Recovery time from Software Standby mode	Crystal resonator connected to main clock oscillator	System clock source is main clock oscillator <sup>*1</sup> MOSCSCR.MOSCSOK P = 0	Enabled	t <sub>SBYMC</sub> <sup>*10</sup>	—	2.33	2.43	ms	Figure 52.23 The division ratio of all oscillators is 1.
		System clock source is main clock oscillator <sup>*1</sup> MOSCSCR.MOSCSOK P = 1	Enabled		—	310	385	μs	
		System clock source is PLL1P with main clock oscillator <sup>*2</sup> MOSCSCR.MOSCSOK P = 0	Enabled	t <sub>SBYPC</sub> <sup>*10</sup>	—	2.47	2.59	ms	
		System clock source is PLL1P with main clock oscillator <sup>*2</sup> MOSCSCR.MOSCSOK P = 1	Enabled		—	388	511	μs	
	External clock input to main clock oscillator	System clock source is main clock oscillator <sup>*3</sup>	Enabled	t <sub>SBYEX</sub> <sup>*10</sup>	—	310	385	μs	
		System clock source is PLL1P with main clock oscillator <sup>*4</sup>	Enabled	t <sub>SBYPE</sub> <sup>*10</sup>	—	388	511	μs	
	System clock source is sub-clock oscillator <sup>*5</sup>		Enabled	t <sub>SBYSC</sub> <sup>*10</sup>	—	0.81	0.87	ms	
	System clock source is HOCO clock oscillator <sup>*6</sup>		Enabled	t <sub>SBYHO</sub> <sup>*10</sup>	—	310	385	μs	
	System clock source is PLL1P with HOCO <sup>*7</sup>		Enabled	t <sub>SBYPH</sub> <sup>*10</sup>	—	398	522	μs	
	System clock source is MOCO clock oscillator <sup>*8</sup>		Enabled	t <sub>SBYMO</sub> <sup>*10</sup>	—	312	387	μs	
Recovery time from Deep Software Standby mode	Deep Software Standby mode 1	Any of PVD0(OFS1(_SEC).PV DLPSEL=1), PVD1, PVD2, or Battery power supply switch is enabled	Standard	t <sub>DSBY</sub>	—	0.68	1.20	ms	Figure 52.24
			Fast		—	0.29	0.62	ms	
		All of PVD0(OFS1(_SEC).PV DLPSEL=1), PVD1, PVD2, and Battery power supply switch are disabled	Standard		—	0.73	1.30		
			Fast		—	0.33	0.71		
	Deep Software Standby mode 2	DPSWCR.WSTS = 0x0B	Standard		—	0.73	1.10	ms	
			Fast		—	0.33	0.50	ms	
		DPSWCR.WSTS = 0x9A	Standard		—	1.60	2.00	ms	
			Fast		—	1.20	1.50	ms	
	Deep Software Standby mode 3		Standard	—	2.10	3.50	ms		
			Fast	—	1.70	2.90	ms		
Wait time after cancellation of Deep Software Standby mode			—	t <sub>DSBYWT</sub>	47.7	—	64.6	μs	

Note 1. When the frequency of the crystal is 48 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x05) and the greatest value of the internal clock division setting is 1.

Note 2. When the frequency of PLL1P is 480 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x05) and the greatest value of the internal clock division setting is 8.

- Note 3. When the frequency of the external clock is 48 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x00) and the greatest value of the internal clock division setting is 1.
- Note 4. When the frequency of PLL1P is 480 MHz (Main Clock Oscillator Wait Control Register (MOSCWTCR) is set to 0x00) and the greatest value of the internal clock division setting is 8.
- Note 5. The Sub-clock oscillator frequency is 32.768 kHz and the greatest value of the internal clock division setting is 1.
- Note 6. The HOCO frequency is 20 MHz and the greatest value of the internal clock division setting is 1.
- Note 7. The PLL frequency is 480 MHz and the greatest value of the internal clock division setting is 8.
- Note 8. The MOCO frequency is 8 MHz and the greatest value of the internal clock division setting is 1.
- Note 9. For details, see SSCR1.SS1FR, DPSBYCR.DCSSMODE bits in the [section 10, Low Power Modes](#).
- Note 10. The recovery time can be calculated with the equation of  $t_{\text{Common}} + \max(t_{\text{OSCSTB}}, t_{\text{PG1}}, t_{\text{PGCK}}) + \max(t_{\text{PG2}}, t_{\text{LPW}})$ . And they can be determined with the following values and equations. For n, the greatest value is selected from among the internal clock(CPUCLK, ICLK, PCLKm, FCLK) division settings (m = A to E).  
 $t_{\text{OSCSTB}}$  in the table below means the time when each oscillator is active. When multiple oscillators are active,  $t_{\text{OSCSTB}}$  is determined by the longest  $t_{\text{OSCSTB}}$  among the active oscillators.
- Note 11. The ICLK frequency is 240 MHz. This recovery time corresponds to  $t_{\text{PG2}}$ .

**Table 52.35 Each element of recovery time**

Wakeup time	Oscillation keep	Fast return function	Typ							Max					Unit
			t <sub>Common</sub>	t <sub>OSCSTB</sub> *1	t <sub>PG1</sub>	t <sub>PGCK</sub>	t <sub>PG2</sub>	t <sub>LPW</sub>	t <sub>Common</sub>	t <sub>OSCSTB</sub> *1	t <sub>PG1</sub>	t <sub>PGCK</sub>	t <sub>PG2</sub>	t <sub>LPW</sub>	
t <sub>SBYMC</sub>	MOSC disabled	Enabled	52.667 + 4/f <sub>ICLK</sub>	t <sub>MAINOSC</sub> WT	75.5	2.1 + 10.5/f <sub>MOCO</sub> + 2.5n/f <sub>MOCO</sub> + 2.5/f <sub>SRCCLK</sub> + 2/f <sub>ICLK</sub>	1449/f <sub>MOCO</sub> + 10/f <sub>ICLK</sub>	10 + 2/f <sub>ICLK</sub> + 2n/f <sub>MAIN</sub>	82.369 + 4/f <sub>ICLK</sub>	t <sub>MAINOSC</sub> WT + 11/0.236	88.8	2.5 + 10.5/f <sub>MOCO</sub> + 2.5n/f <sub>MOCO</sub> + 2.5/f <sub>SRCCLK</sub> + 2/f <sub>ICLK</sub>	1449/f <sub>MOCO</sub> + 10/f <sub>ICLK</sub>	10 + 2/f <sub>ICLK</sub> + 2n/f <sub>MAIN</sub>	μs
	MOSC enabled	Enabled	52.667 + 4/f <sub>ICLK</sub>	3/0.262				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>MAIN</sub>	82.369 + 4/f <sub>ICLK</sub>	14/0.236				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>MAIN</sub>	
t <sub>SBYPC</sub>	MOSC disabled	Enabled	52.667 + 4/f <sub>ICLK</sub>	24.125 + t <sub>MAINOSC</sub> CWT + 31/0.262*2				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>PLL</sub>	82.369 + 4/f <sub>ICLK</sub>	24.05 + t <sub>MAINOSC</sub> WT + 42/0.236*3				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>PLL</sub>	μs
	MOSC enabled	Enabled	52.667 + 4/f <sub>ICLK</sub>	24.125 + 34/0.262*2				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>PLL</sub>	82.369 + 4/f <sub>ICLK</sub>	24.05 + 45/0.236*3				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>PLL</sub>	
t <sub>SBYEX</sub>	—	Enabled	52.667 + 4/f <sub>ICLK</sub>	3/0.262				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>EXMAIN</sub>	82.369 + 4/f <sub>ICLK</sub>	14/0.236				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>EXMAIN</sub>	μs
t <sub>SBYPE</sub>	—	Enabled	52.667 + 4/f <sub>ICLK</sub>	24.125 + 34/0.262*2				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>PLL</sub>	82.369 + 4/f <sub>ICLK</sub>	24.05 + 45/0.236*3				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>PLL</sub>	μs
t <sub>SBYSC</sub>	—	Enabled	52.667 + 4/f <sub>ICLK</sub>	0				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>SOSC</sub>	82.369 + 4/f <sub>ICLK</sub>	0				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>SOSC</sub>	μs
t <sub>SBYHO</sub>	—	Enabled	52.667 + 4/f <sub>ICLK</sub>	23.375				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>HOCO</sub>	82.369 + 4/f <sub>ICLK</sub>	70.234				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>HOCO</sub>	μs
t <sub>SBYPH</sub>	—	Enabled	52.667 + 4/f <sub>ICLK</sub>	24.125 + 140*2				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>PLL</sub>	82.369 + 4/f <sub>ICLK</sub>	24.05 + 202*3				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>PLL</sub>	μs
t <sub>SBYMO</sub>	—	Enabled	52.667 + 4/f <sub>ICLK</sub>	0				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>MOCO</sub>	82.369 + 4/f <sub>ICLK</sub>	0				10 + 2/f <sub>ICLK</sub> + 2n/f <sub>MOCO</sub>	μs

Note: The unit of frequency is MHz.

Note 1. If more than one oscillator is operating, the largest value of the operating oscillator in this column is applied.

Note 2. "24.125" can be reduced when both PLL1LDOCR.SKEEP and PLL2LDOCR.SKEEP are 1.

Note 3. "24.05" can be reduced when both PLL1LDOCR.SKEEP and PLL2LDOCR.SKEEP are 1.

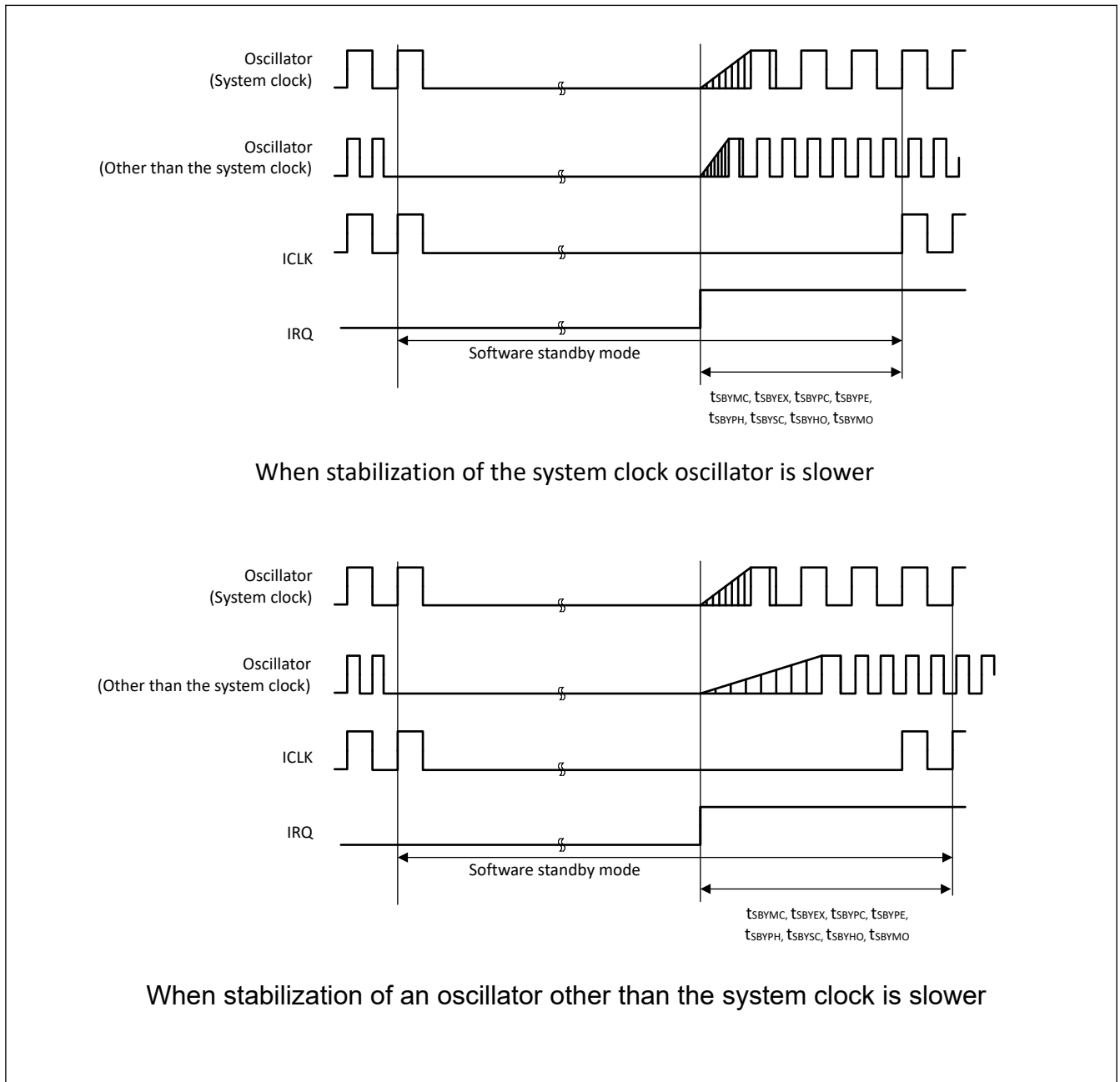


Figure 52.23 Software Standby mode cancellation timing



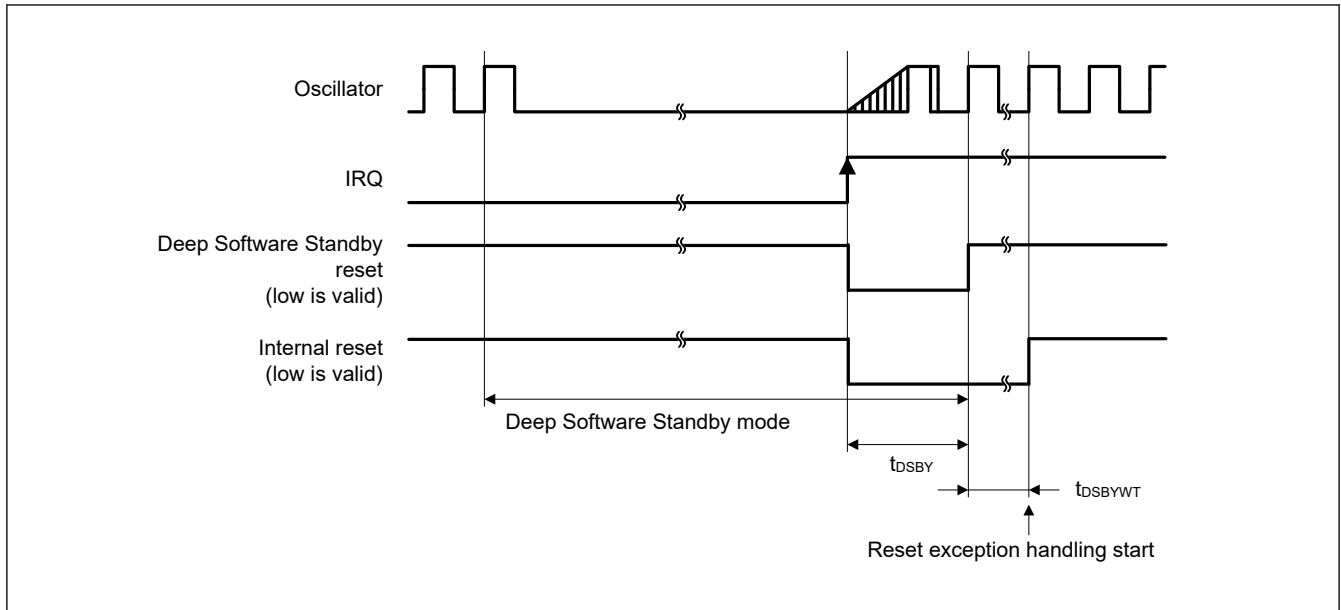


Figure 52.24 Deep Software Standby mode cancellation timing

### 52.3.5 NMI and IRQ Noise Filter

Table 52.36 NMI and IRQ noise filter

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions	
NMI pulse width	$t_{NMIW}$	200	—	—	ns	NMI digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2^{*1}$	—	—			$t_{Pcyc} \times 2 > 200$ ns
	200	—	—	NMI digital filter enabled		$t_{NMICK} \times 3 \leq 200$ ns	
						$t_{NMICK} \times 3.5^{*2}$	—
IRQ pulse width	$t_{IRQW}$	200	—	—	ns	IRQ digital filter disabled	$t_{Pcyc} \times 2 \leq 200$ ns
		$t_{Pcyc} \times 2^{*1}$	—	—			$t_{Pcyc} \times 2 > 200$ ns
	200	—	—	IRQ digital filter enabled		$t_{IRQCK} \times 3 \leq 200$ ns	
						$t_{IRQCK} \times 3.5^{*3}$	—

- Note: 200 ns minimum in Software Standby mode.
- Note: If the system clock source is switched, add 4 clock cycles of the switched source.
- Note 1.  $t_{Pcyc}$  indicates the PCLKB cycle.
- Note 2.  $t_{NMICK}$  indicates the cycle of the NMI digital filter sampling clock.
- Note 3.  $t_{IRQCK}$  indicates the cycle of the IRQi digital filter sampling clock.

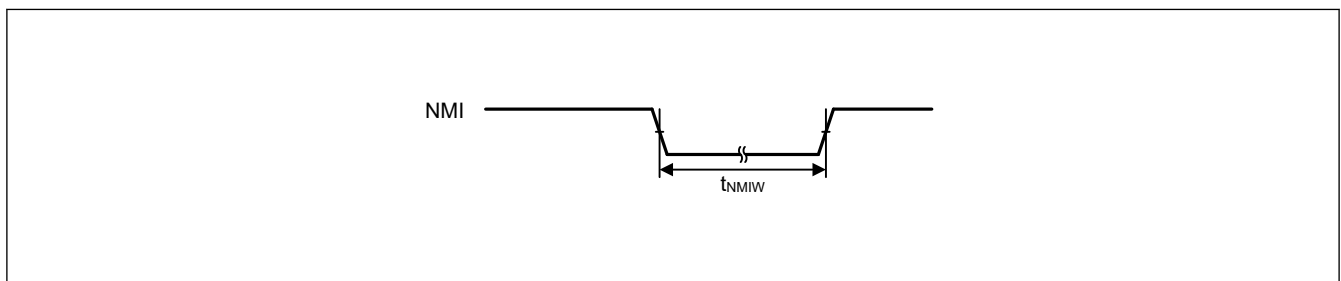


Figure 52.25 NMI interrupt input timing

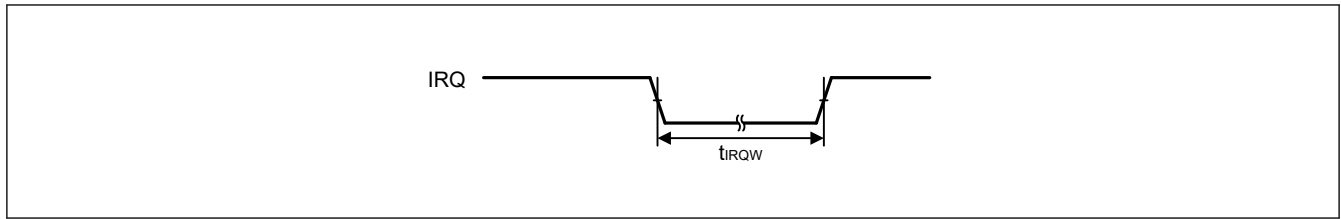


Figure 52.26 IRQ interrupt input timing

### 52.3.6 I/O Ports, POEG, GPT, AGT, ULPT and ADC12 Trigger Timing

Table 52.37 I/O ports, POEG, GPT, AGT, ULPT and ADC12 trigger timing (1 of 2)

GPT Conditions:

Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

If GPT pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

AGT Conditions:

Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Test conditions		
I/O ports	Input data pulse width	$t_{PRW}$	5.5	—	$t_{cyc}$	Figure 52.27		
	EXCIN input frequency	$t_{EXCIN}$	—	36	kHz			
	RTCICn (n = 0 to 2) input pulse width	$t_{RTCICW}$	13.89	—	$\mu s$	Figure 52.28		
POEG	POEG input trigger pulse width	$t_{POEW}$	3	—	$t_{Pcyc}$	Figure 52.29		
GPT	Input capture pulse width (Cycle)	Single edge	$t_{GTICW}^{*1}$	1.5	—	$t_{PDcyc}$	Figure 52.30	
		Dual edge		2.5	—			
	Input capture pulse width (Time)	2.70V or above	$t_{GTICW}^{*1}$	12.5	—	ns		
		1.68V or above (VCC)		25.0	—			
		1.65V or above (VCC2)						
	GTIOCxY output skew (x = 0 to 5, Y = A or B)	2.70V or above	$t_{GTISK}$	—	4	ns		Figure 52.31
		1.68V or above (VCC)		—	5			
	GTIOCxY output skew (x = 10 to 13, Y = A or B)	2.70V or above	$t_{GTISK}$	—	4	ns		
		1.68V or above (VCC)		—	5			
	GTIOCxY output skew (x = 0 to 5, 10 to 13, Y = A or B)	2.70V or above	$t_{GTISK}$	—	6	ns		
1.68V or above (VCC)		—		7				
	1.65V or above (VCC2)							

**Table 52.37 I/O ports, POEG, GPT, AGT, ULPT and ADC12 trigger timing (2 of 2)**

GPT Conditions:

Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

If GPT pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

AGT Conditions:

Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter		Symbol	Min	Max	Unit	Test conditions	
AGT	AGTIO, AGTEE input cycle	2.70V or above	100	—	ns	Figure 52.32	
		1.68V or above (VCC) 1.65V or above (VCC2)	100	—			
	AGTIO, AGTEE input high width, low width	2.70V or above	$t_{ACKWH}, t_{ACKWL}$	40	—		ns
		1.68V or above (VCC) 1.65V or above (VCC2)	40	—			
	AGTIO, AGTO, AGTOA, AGTOB output cycle	2.70V or above	$t_{ACYC2}$	62.5	—		ns
		1.68V or above (VCC) 1.65V or above (VCC2)	62.5	—			
ULPT	ULPTEE, ULPTEVI input cycle	2.70V or above	32	-	$\mu$ s	Figure 52.33	
		1.68V or above (VCC) 1.65V or above (VCC2)	32	-			
	ULPTEE, ULPTVI input high width, low width	2.70V or above	$t_{ULCKWH}, t_{ULCKWL}$	12	-		$\mu$ s
		1.68V or above (VCC) 1.65V or above (VCC2)	12	-			
	ULPTO, ULPTOA, ULPTOB output cycle	2.70V or above	$t_{ULCYC2}$	64	-		$\mu$ s
		1.68V or above (VCC) 1.65V or above (VCC2)	64	-			
ADC12	ADC12 trigger input pulse width	2.70V or above	1.5	—	$t_{Pcyc}$	Figure 52.34	
		1.68V or above (VCC) 1.65V or above (VCC2)	3.0	—			

Note:  $t_{cyc}$ : ICLK cycle,  $t_{Pcyc}$ : PCLKB cycle,  $t_{PDcyc}$ : PCLKD cycle,  $t_{ULPTLCLK}$ : ULPTLCLK cycle..

Note 1. For Cycle and Time, the longer time characteristics is applied.

Note 2. Constraints on input cycle:

When not switching the source clock:  $t_{Pcyc} \times 2 < t_{ACYC}$  should be satisfied.When switching the source clock:  $t_{Pcyc} \times 6 < t_{ACYC}$  should be satisfied.

Note 3. Constraints on input cycle:

ULPTEVI :  $t_{Pcyc} \times 2 < t_{ULCYC}$  should be satisfied.ULPTEE:  $t_{ULPTLCLK} \times 2 < t_{ULCYC}$  should be satisfied.

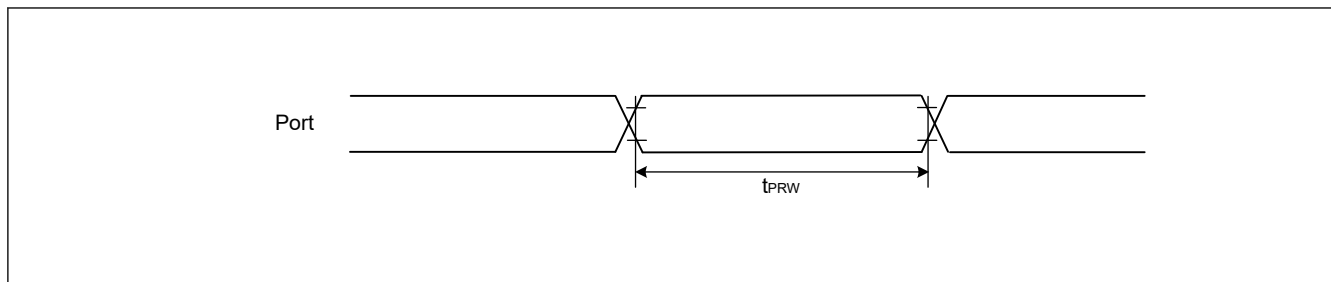


Figure 52.27 I/O ports input timing

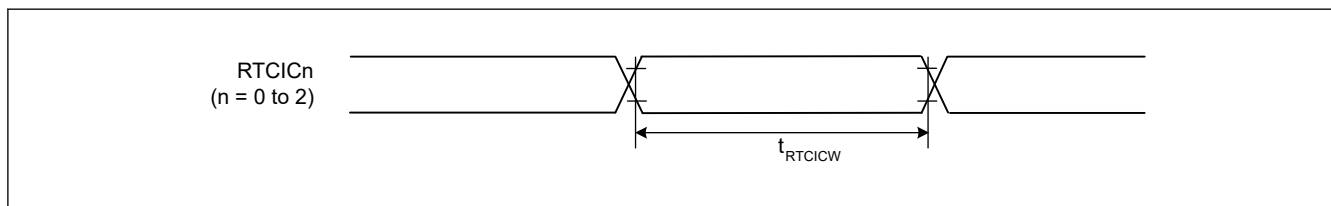


Figure 52.28 RTCICn input timing

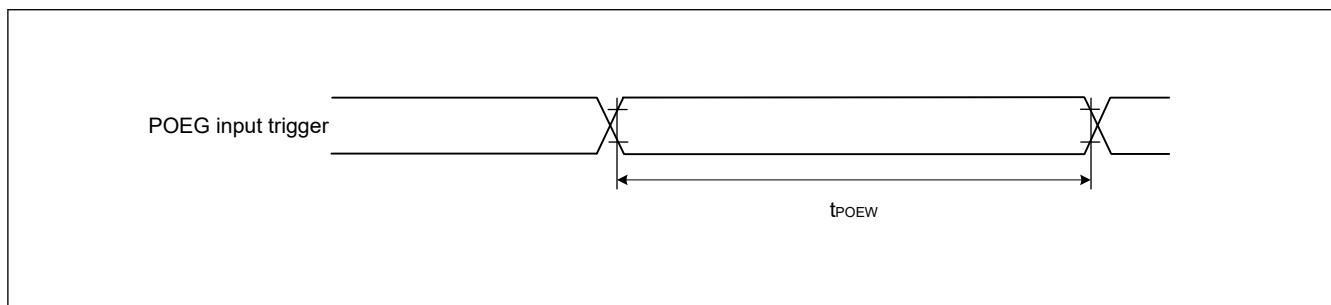


Figure 52.29 POEG input trigger timing

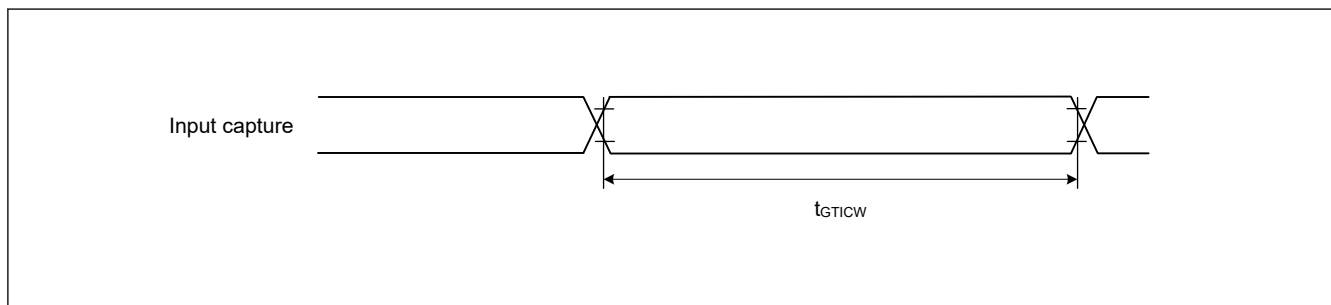


Figure 52.30 GPT input capture timing

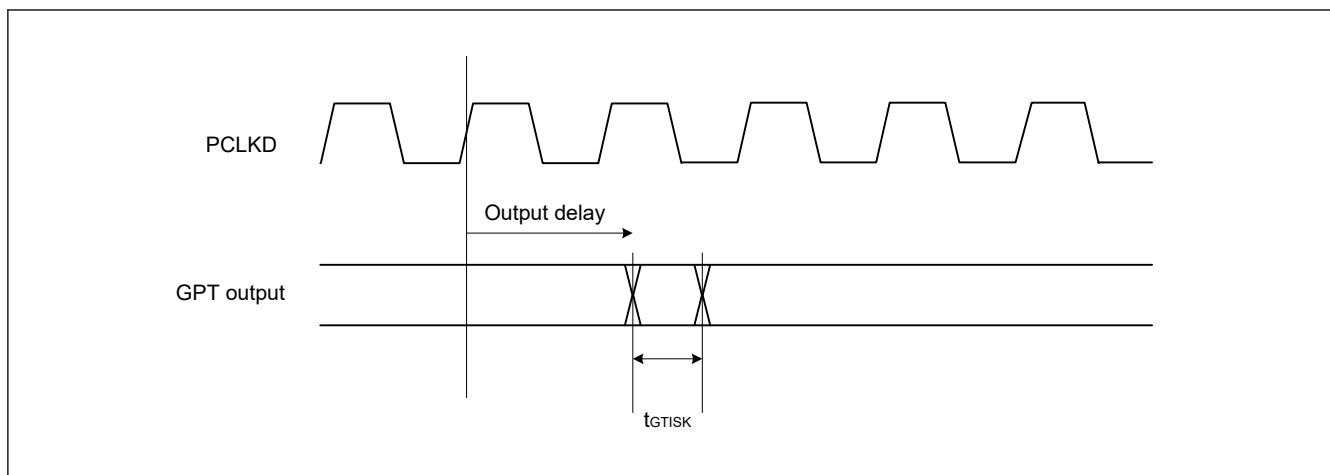


Figure 52.31 GPT output delay skew

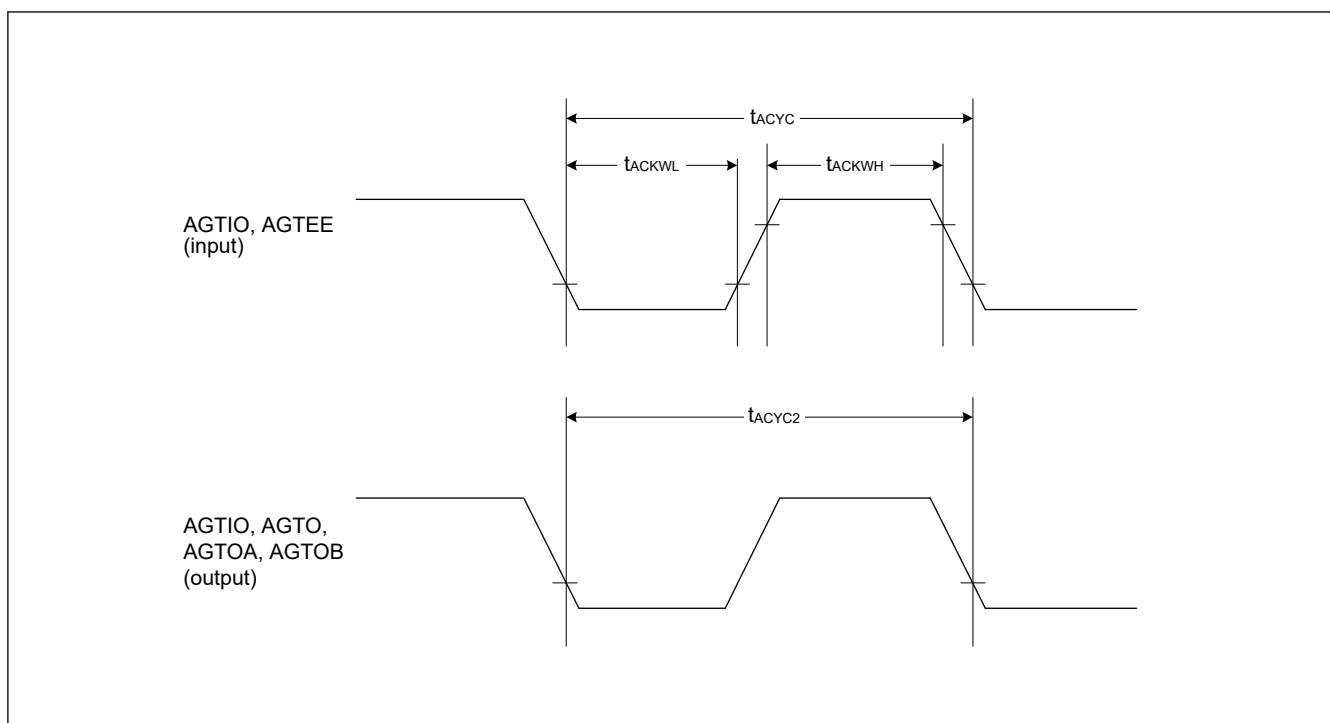


Figure 52.32 AGT input/output timing

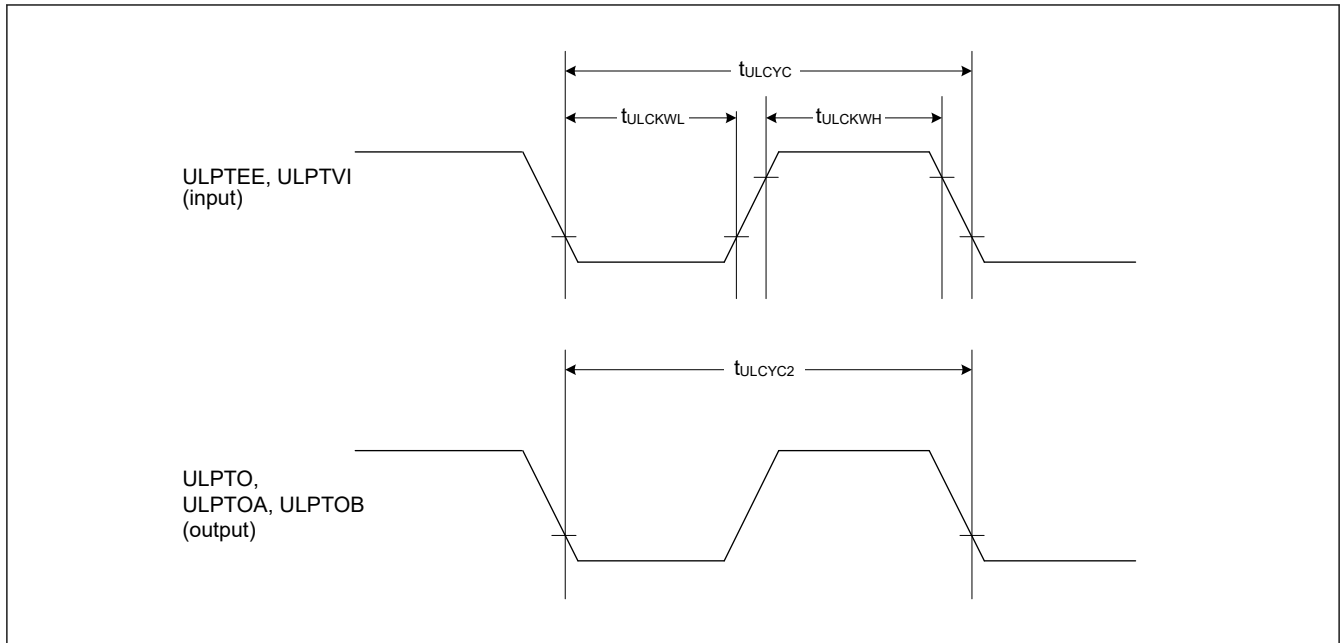


Figure 52.33 ULPT input/output timing

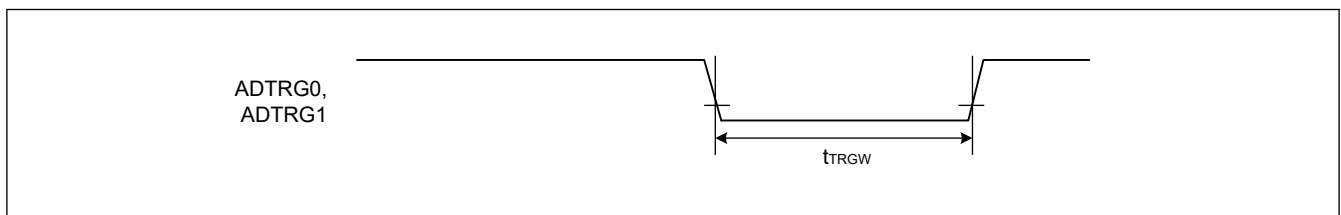


Figure 52.34 ADC12 trigger input timing

### 52.3.7 CAC Timing

Table 52.38 CAC timing

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
CAC CACREF input pulse width	$t_{CACREF}$	$t_{PBcyc} \leq t_{cac}^{*1}$	$4.5 \times t_{cac} + 3 \times t_{PBcyc}$	—	—	ns
		$t_{PBcyc} > t_{cac}^{*1}$	$5 \times t_{cac} + 6.5 \times t_{PBcyc}$	—	—	ns

Note:  $t_{PBcyc}$ : PCLKB cycle.

Note 1.  $t_{cac}$ : CAC count clock source cycle.

### 52.3.8 SCI Timing

**Table 52.39 SCI timing (Asynchronous mode)**

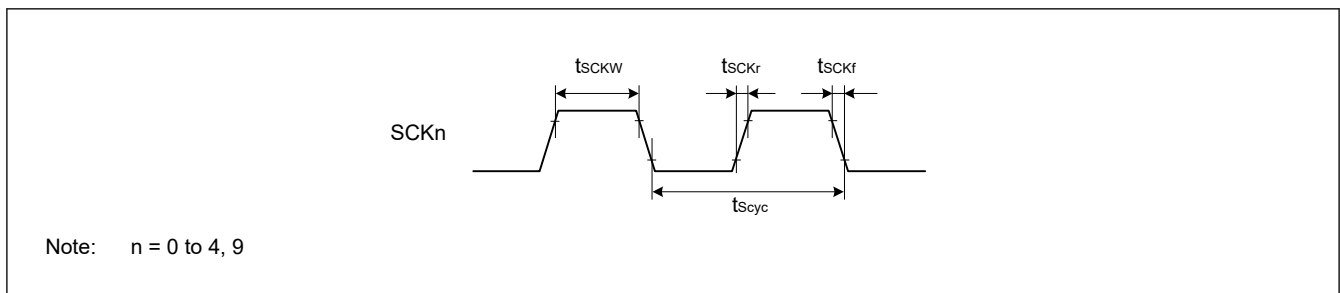
Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter	VCC/VCC2	Symbol	Min	Max	Unit	Note
Input clock cycle	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{Scyc}$	4.0	—	$t_{Tcyc}$	Figure 52.35
Input clock pulse width	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{SCKW}$	0.4	—	$t_{Scyc}$	
Input clock rise time	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{SCKr}$	—	0.1 <sup>*1</sup>	$t_{Scyc}$	
Input clock fall time	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{SCKf}$	—	0.1 <sup>*1</sup>	$t_{Scyc}$	
Output clock cycle	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{Scyc}$	6.0	—	$t_{Tcyc}$	
Output clock pulse width	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{SCKW}$	0.4	—	$t_{Scyc}$	
Output clock rise time	2.70 V or above	$t_{SCKr}$	—	3.3	ns	
	1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.6		
Output clock fall time	2.70 V or above	$t_{SCKf}$	—	3.3	ns	
	1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.6		

Note:  $t_{Tcyc}$ : TCLK cycle.

Note 1. 1  $\mu$ s at the longest



**Figure 52.35 SCK clock input/output timing**

**Table 52.40 SCI timing (Simple SPI) (1 of 2)**

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter		High Speed/Default	VCC/VCC2	Symbol	Min	Max	Unit	Note	
SCK clock cycle output	Master	—	2.70 V or above	$t_{SPcyc}$	2.0	65536	$t_{Cyc}$	Figure 52.36	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	65536			
SCK clock cycle input	Slave	—	2.70 V or above	$t_{SPcyc}$	2.0	65536	$t_{Cyc}$		Figure 52.36
			1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	65536			
SCK clock high pulse width	Master	—	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{SPCKWH}$	0.4	—	$t_{SPcyc}$		
	Slave	—							
SCK clock low pulse width	Master	—	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{SPCKWL}$	0.4	—	$t_{SPcyc}$		
	Slave	—							
SCK clock rise and fall time	Output	—	2.70 V or above	$t_{SPCKr},$ $t_{SPCKf}$	—	3.3	ns		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.6			
	Input	—	2.70 V or above		—	$0.1^{*3}$	$t_{SPcyc}$		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	$0.1^{*3}$			
Data input setup time	Master	High Speed <sup>*1</sup>	2.70 V or above	$t_{SU}$	14.9 - (AST[2:0] settings)	—	ns	Figure 52.37, Figure 52.38	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		23.1 - (AST[2:0] settings)	—			
		Default <sup>*2</sup>	2.70 V or above		16.2 - (AST[2:0] settings)	—			
			1.68 V or above (VCC) 1.65 V or above (VCC2)		23.8 - (AST[2:0] settings)	—			
	Slave	Default <sup>*2</sup>	2.70 V or above		2.5	—			
			1.68 V or above (VCC) 1.65 V or above (VCC2)		4.5	—			
Data input hold time	Master	High Speed <sup>*1</sup>	2.70 V or above	$t_H$	-3.2 + (AST[2:0] settings)	—	ns	Figure 52.37, Figure 52.38	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-3.2 + (AST[2:0] settings)	—			
		Default <sup>*2</sup>	2.70 V or above		-3.2 + (AST[2:0] settings)	—			
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-3.2 + (AST[2:0] settings)	—			
	Slave	Default <sup>*2</sup>	2.70 V or above		2.5	—			
			1.68 V or above (VCC) 1.65 V or above (VCC2)		4.5	—			



**Table 52.40 SCI timing (Simple SPI) (2 of 2)**

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter		High Speed/Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
Data output delay	Master	High Speed*1	2.70 V or above	$t_{OD}$	—	3.0	ns	Figure 52.37, Figure 52.38
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	4.5		
		Default*2	2.70 V or above		—	3.5		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	5.5		
	Slave	High Speed*1	2.70 V or above		—	15.0		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	23.0		
Default*2	2.70 V or above	—	21.0					
	1.68 V or above (VCC) 1.65 V or above (VCC2)	—	29.0					
Data output hold time	Master	High Speed*1	2.70 V or above	$t_{OH}$	-3.0	—	ns	Figure 52.37, Figure 52.38
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-4.5	—		
		Default*2	2.70 V or above		-3.5	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-5.5	—		
	Slave	Default*2	2.70 V or above		0.0	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		0.0	—		
Data rise and fall time	Output	—	2.70 V or above	$t_{Dr}, t_{Df}$	—	3.3	ns	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.6		
	Input	—	2.70 V or above		—	1.0		$\mu$ s
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	1.0		
SS input setup time	—	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{LEAD}$	1.0	—	$t_{SPCyc}$	Figure 52.39, Figure 52.40	
SS input hold time	—	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{LAG}$	1.0	—	$t_{SPCyc}$		
SS input rise and fall time	—	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{SSLr}, t_{SSLf}$	—	1.0	$\mu$ s	—	
Slave access time	—	2.70 V or above 1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{SA}$	—	$3 \times t_{TCyc} + 25$	ns	Figure 52.39, Figure 52.40	
				—	$3 \times t_{TCyc} + 32$			
Slave output release time	—	2.70 V or above 1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{REL}$	—	$3 \times t_{TCyc} + 25$	ns		
				—	$3 \times t_{TCyc} + 32$			

Note:  $t_{TCyc}$ : TCLK cycle.

Note 1. Must use pins that have a letter appended to their name, for instance \_A, \_B, \_C, to indicate group membership. SCI0, SCI1, SCI2, SCI3 and SCI9 are instance \_A, SCI4 is instance \_B.

Note 2. All pins of group membership can be used.

Note 3. 1  $\mu$ s at the longest

**Table 52.41 SCI timing (Clock synchronous mode) (1 of 3)**

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter		High Speed/Default	VCC/VCC2	Symbol	Min	Max	Unit	Note			
SCK clock cycle output	Master	—	2.70 V or above	$t_{Scyc}$	2.0	—	$t_{Scyc}$				
			1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	—					
SCK clock cycle input	Slave	—	2.70 V or above		2.0	—					
			1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	—					
SCK clock high pulse width	Master	—	2.70 V or above		$t_{SCKWH}$	0.4			—	$t_{Scyc}$	
	Slave		1.68 V or above (VCC) 1.65 V or above (VCC2)								
SCK clock low pulse width	Master	—	2.70 V or above	$t_{SCKWL}$	0.4	—	$t_{Scyc}$				
	Slave		1.68 V or above (VCC) 1.65 V or above (VCC2)								
SCK clock rise and fall time	Output	—	2.70 V or above	$t_{SCKr, SCKf}$	—	3.3	ns				
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.6					
	Input	—	1.68 V or above (VCC) 1.65 V or above (VCC2)		—	0.1 <sup>*3</sup>	$t_{Scyc}$				
Data input setup time	Master	High Speed <sup>*1</sup>	2.70 V or above	$t_{SU}$	15.1 - (AST[2:0] settings)	—	ns				
			1.68 V or above (VCC) 1.65 V or above (VCC2)		23.2 - (AST[2:0] settings)	—					
		Default <sup>*2</sup>	2.70 V or above		16.5 - (AST[2:0] settings)	—					
			1.68 V or above (VCC) 1.65 V or above (VCC2)		24.2 - (AST[2:0] settings)	—					
	Slave	Default <sup>*2</sup>	2.70 V or above		3.3	—					
			1.68 V or above (VCC) 1.65 V or above (VCC2)		5.3	—					

**Table 52.41 SCI timing (Clock synchronous mode) (2 of 3)**

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter		High Speed/Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
Data input hold time	Master	High Speed*1	2.70 V or above	$t_H$	-3.3 + (AST[2:0] settings)	—	ns	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-3.3 + (AST[2:0] settings)	—		
		Default*2	2.70 V or above		-3.2 + (AST[2:0] settings)	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-3.2 + (AST[2:0] settings)	—		
	Slave	Default*2	2.70 V or above		3.0	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		5.0	—		
Data output delay	Master	High Speed*1	2.70 V or above	$t_{OD}$	—	5.0	ns	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	5.0		
		Default*2	2.70 V or above		—	7.3		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	7.3		
	Slave	High Speed*1	2.70 V or above		—	15.0		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	23.0		
		Default*2	2.70 V or above		—	21.0		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		—	29.0		

**Table 52.41 SCI timing (Clock synchronous mode) (3 of 3)**

Conditions: High drive output is selected in the Port Drive Capability bit in the PmnPFS register.

If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter		High Speed/Default	VCC/VCC2	Symbol	Min	Max	Unit	Note
Data output hold time	Master	High Speed*1	2.70 V or above	$t_{OH}$	-5.0	—	ns	
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-5.0	—		
		Default*2	2.70 V or above		-7.3	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		-7.3	—		
	Slave	High Speed*1	2.70 V or above		0	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		0	—		
		Default*2	2.70 V or above		0	—		
			1.68 V or above (VCC) 1.65 V or above (VCC2)		0	—		
Data rise and fall time	Output	—	2.70 V or above	$t_{Dr}, t_{Df}$	—	3.3	ns	
		—	1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.6		
	Input	—	1.68 V or above (VCC) 1.65 V or above (VCC2)		—	1.0		

Note:  $t_{Toc}$ : TCLK cycle.

Note 1. Must use pins that have a letter appended to their name, for instance \_A, \_B, \_C, to indicate group membership. SCI0, SCI1, SCI2, SCI3 and SCI9 are instance \_A, SCI4 is instance \_B.

Note 2. All pins of group membership can be used.

Note 3. 1  $\mu$ s at the longest

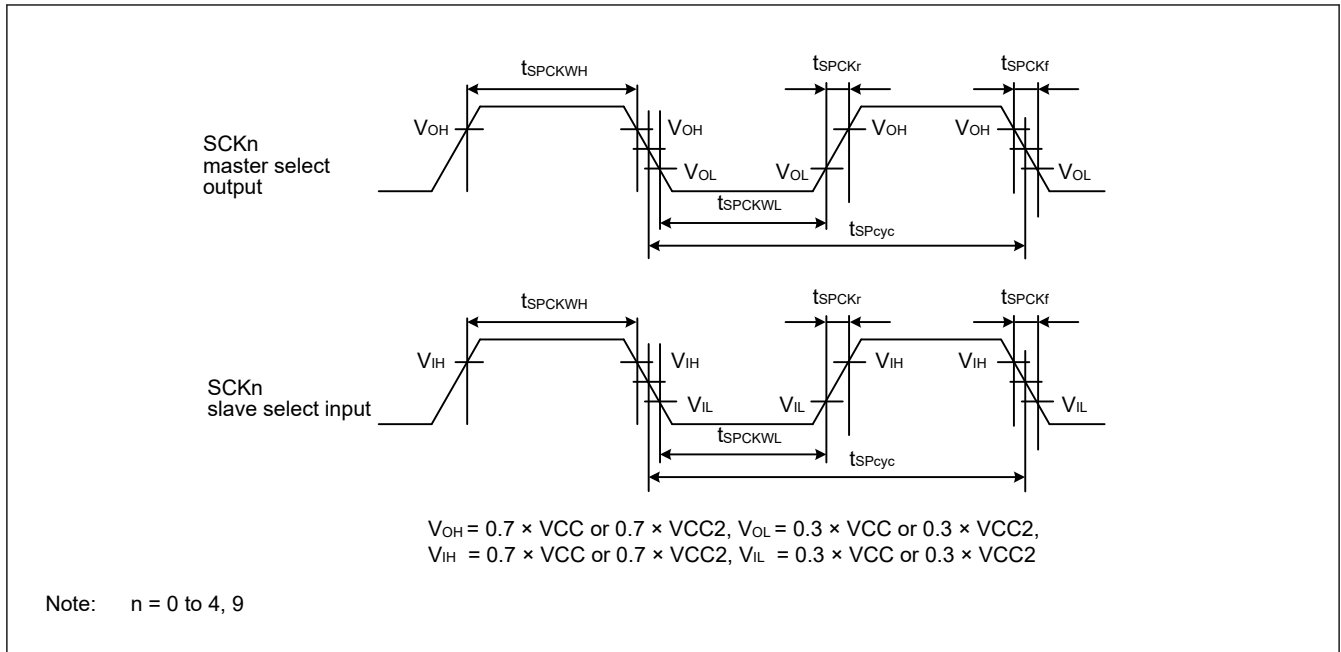


Figure 52.36 SCI simple SPI mode clock timing

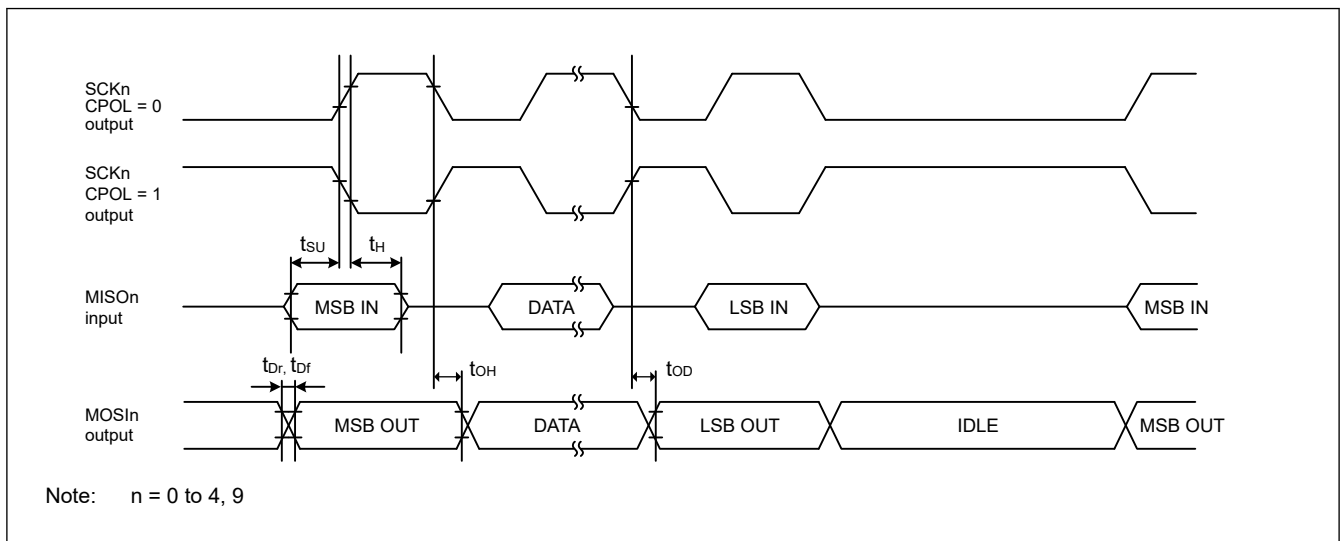


Figure 52.37 SCI simple SPI mode timing for master when CPHA = 0

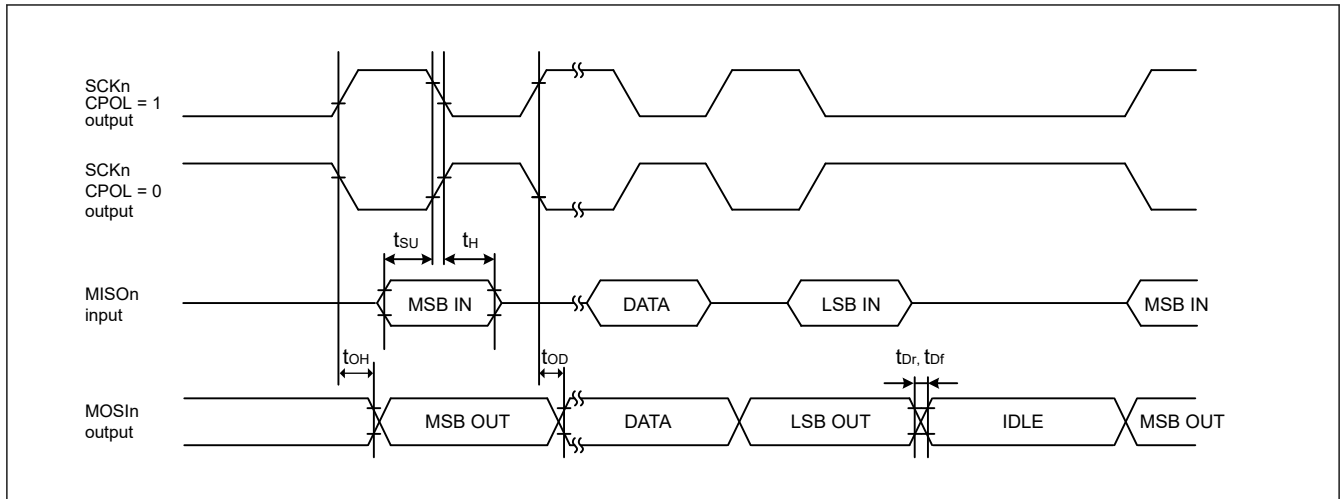


Figure 52.38 SCI simple SPI mode timing for master when CPHA = 1

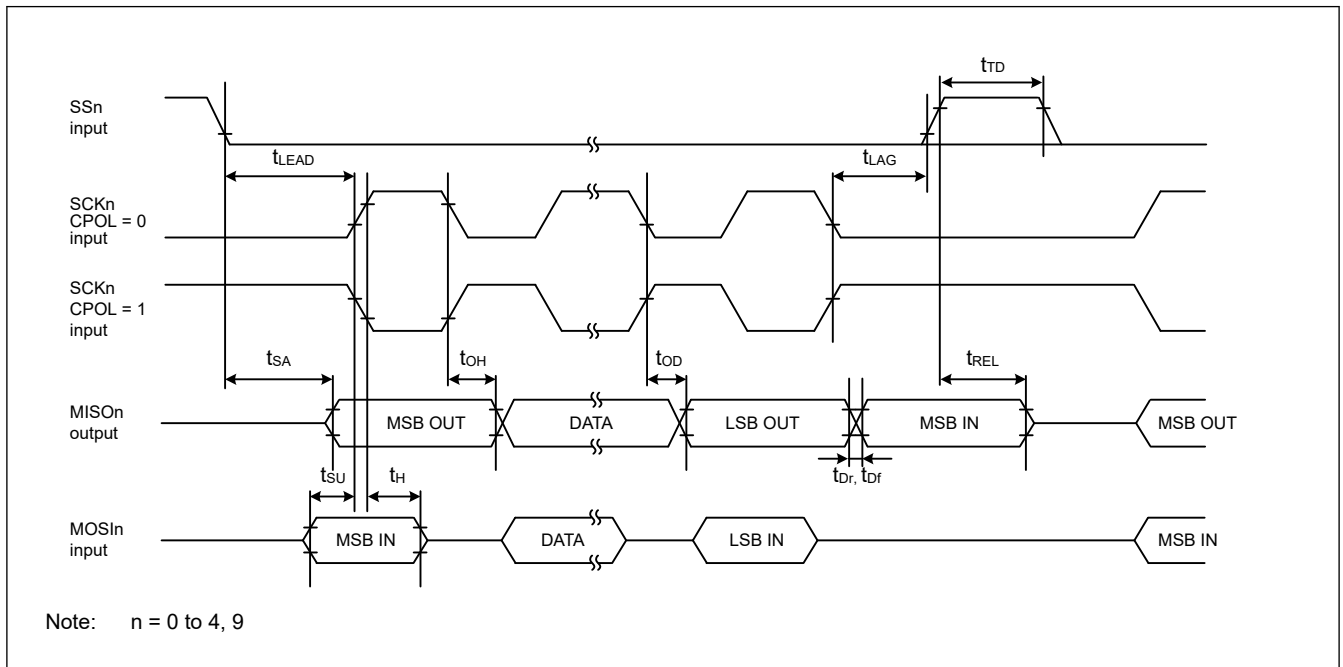


Figure 52.39 SCI simple SPI mode timing for slave when CPHA = 0

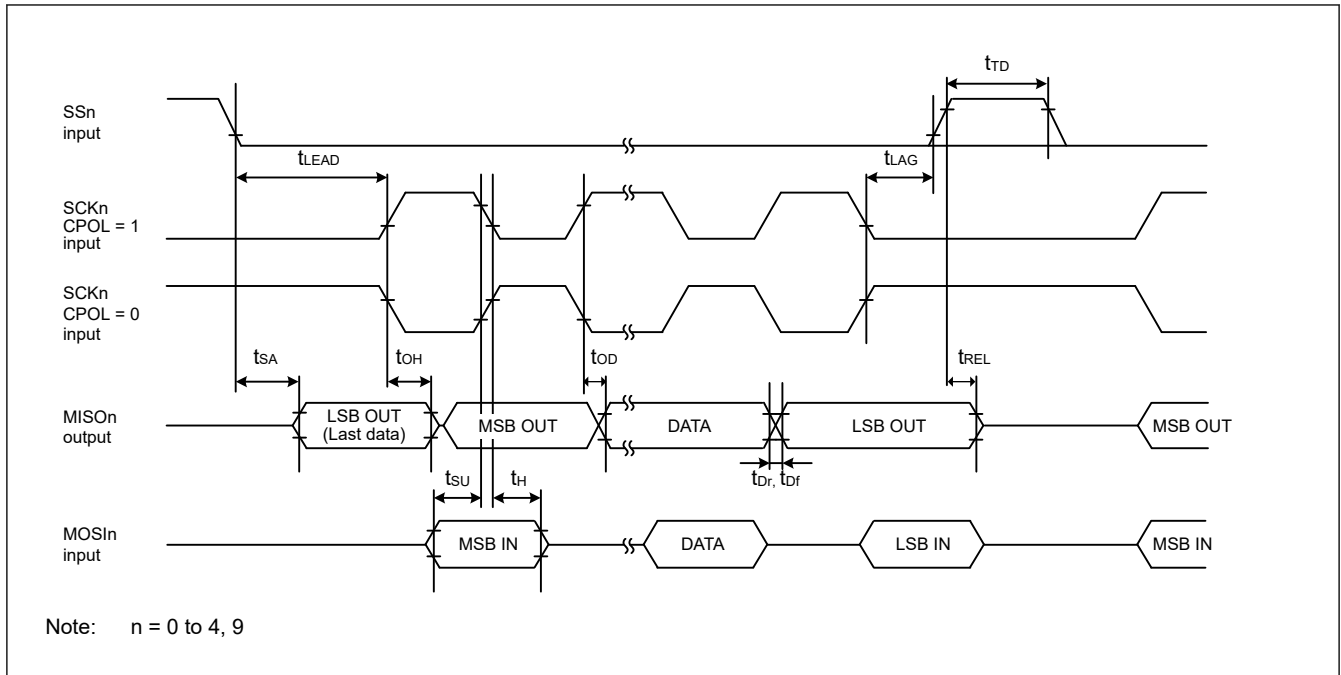


Figure 52.40 SCI simple SPI mode timing for slave when CPHA = 1

Table 52.42 SCI timing (Simple IIC mode)

Conditions: Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

VCC: 1.68V or above, VCC2: 1.65V or above

If SCI pins are specified across the VCC I/O and VCC2 I/O, characteristics below is guaranteed only when VCC = VCC2.

Parameter	Symbol	Min	Max	Unit	Note
Simple IIC (Standard mode)	SCL, SDA input rise time	$t_{sr}$	—	1000	ns
	SCL, SDA input fall time	$t_{sf}$	—	300	ns
	SCL, SDA input spike pulse removal time	$t_{sp}$	0	$4 \times t_{Tcyc}$	ns
	Data input setup time	$t_{SDAS}$	250	—	ns
	Data input hold time	$t_{SDAH}$	0	—	ns
	SCL, SDA capacitive load	$C_b^{*1}$	—	400	pF
Simple IIC (Fast mode)	SCL, SDA input rise time	$t_{sr}$	—	300	ns
	SCL, SDA input fall time	$t_{sf}$	—	300	ns
	SCL, SDA input spike pulse removal time	$t_{sp}$	0	$4 \times t_{Tcyc}$	ns
	Data input setup time	$t_{SDAS}$	100	—	ns
	Data input hold time	$t_{SDAH}$	0	—	ns
	SCL, SDA capacitive load	$C_b^{*1}$	—	400	pF

Note:  $t_{Tcyc}$ : TCLK cycle.

Note 1.  $C_b$  indicates the total capacity of the bus line.

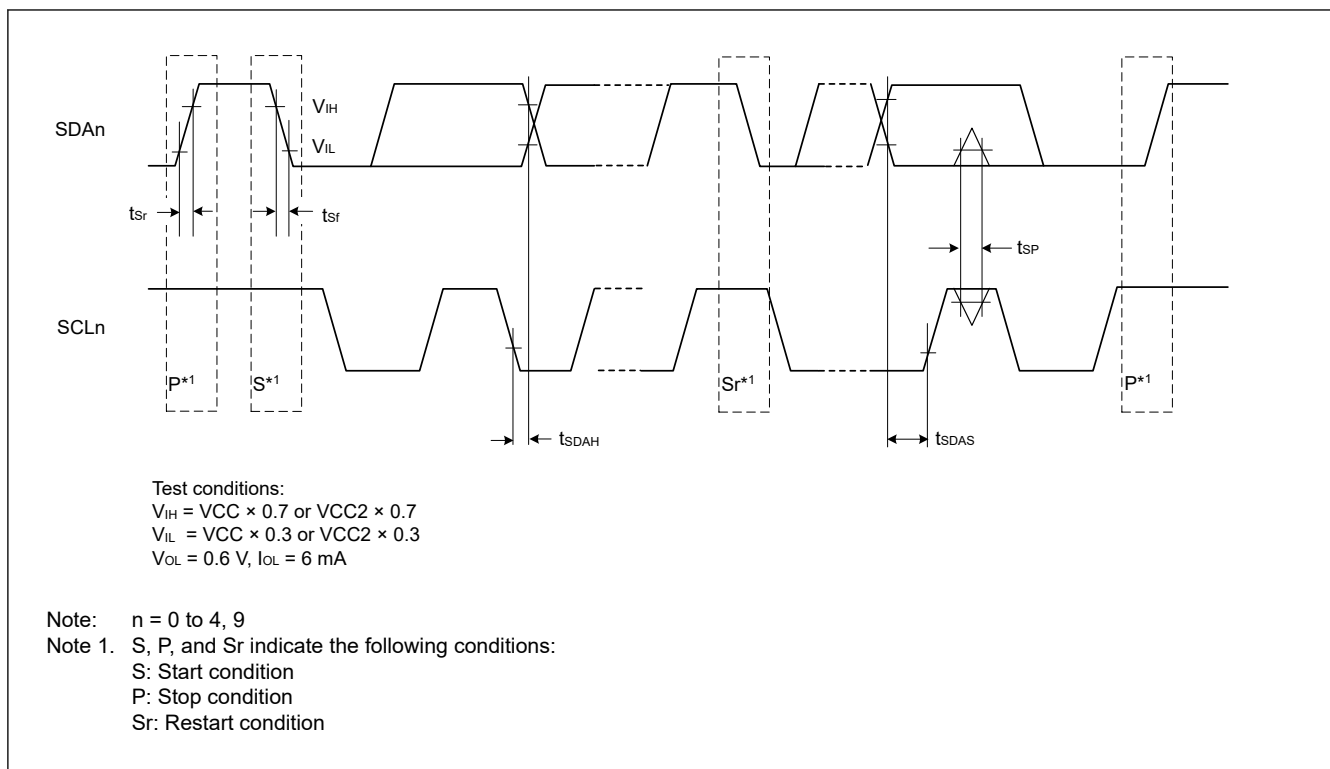


Figure 52.41 SCI simple IIC mode timing



## 52.3.9 SPI Timing

**Table 52.43 SPI timing (1 of 4)**

Conditions:

1. High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
2. Use pins that have a letter appended to their names, for instance "\_A" or "\_B" to indicate group membership.
3. Load capacitance C = 15pF is applied to the VCC/VCC2 condition "3.00 V or above".

Parameter	VCC/VCC2	Symbol	Min	Max	Unit	Note	
RSPCK clock cycle	Master	3.00 V or above	$t_{SPCyc}$	2.0	4096	$t_{Tcyc}$	Figure 52.42
		2.70 V or above		2.0	4096		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	4096		
	Slave	3.00 V or above		2.0	—		
		2.70 V or above		2.0	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	—		
RSPCK clock high pulse width	Master	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{SPCKWH}$	$(t_{SPCyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$		ns	
	Slave	1.68 V or above (VCC) 1.65 V or above (VCC2)		0.4	—	$t_{SPCyc}$	
RSPCK clock low pulse width	Master	1.68 V or above (VCC) 1.65 V or above (VCC2)	$t_{SPCKWL}$	$(t_{SPCyc} - t_{SPCKr} - t_{SPCKf}) / 2 - 3$		ns	
	Slave	1.68 V or above (VCC) 1.65 V or above (VCC2)		0.4	—	$t_{SPCyc}$	
RSPCK clock rise and fall time	Output	3.00 V or above	$t_{SPCKr}, t_{SPCKf}$	—	1.66	ns	
		2.70 V or above		—	3.30		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.60		
	Input	3.00 V or above		—	0.1*1	$t_{SPCyc}$	
		2.70 V or above		—	0.1*1		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	0.1*1		

**Table 52.43 SPI timing (2 of 4)**

Conditions:

1. High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
2. Use pins that have a letter appended to their names, for instance "\_A" or "\_B" to indicate group membership.
3. Load capacitance C = 15pF is applied to the VCC/VCC2 condition "3.00 V or above".

Parameter		VCC/VCC2	Symbol	Min	Max	Unit	Note
Data input setup time	Master	3.00 V or above	$t_{SU}$	-2.5	—	ns	Figure 52.43, Figure 52.44
		2.70 V or above		0.0	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		0.0	—		
	Slave	3.00 V or above		2.5	—		
		2.70 V or above		2.5	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		2.5	—		
Data input hold time	Master	3.00 V or above	$t_H$	7.5	—	ns	
		2.70 V or above		7.5	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		9.5	—		
	Slave	3.00 V or above		2.5	—		
		2.70 V or above		2.5	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		5.5	—		
SSL setup time	Master	3.00 V or above	$t_{LEAD}$	$1 \times t_{SPcyc} - 10$	$8 \times t_{SPcyc} + 10$	ns	
		2.70 V or above		$1 \times t_{SPcyc} - 10$	$8 \times t_{SPcyc} + 10$		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		$1 \times t_{SPcyc} - 10$	$8 \times t_{SPcyc} + 10$		
	Slave	3.00 V or above		4.0	—	$t_{Tcyc}$	
		2.70 V or above		4.0	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	—		
SSL hold time	Master	3.00 V or above	$t_{LAG}$	$1 \times t_{SPcyc} - 10$	$8 \times t_{SPcyc} + 10$	ns	
		2.70 V or above		$1 \times t_{SPcyc} - 10$	$8 \times t_{SPcyc} + 10$		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		$1 \times t_{SPcyc} - 10$	$8 \times t_{SPcyc} + 10$		
	Slave	3.00 V or above		4.0	—	$t_{Tcyc}$	
		2.70 V or above		4.0	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		4.0	—		

**Table 52.43 SPI timing (3 of 4)**

Conditions:

- High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
- Use pins that have a letter appended to their names, for instance "\_A" or "\_B" to indicate group membership.
- Load capacitance C = 15pF is applied to the VCC/VCC2 condition "3.00 V or above".

Parameter	VCC/VCC2	Symbol	Min	Max	Unit	Note		
TI SSP SS input setup time	Slave	3.00 V or above	$t_{TISS}$	2.5	—	ns	Figure 52.48	
				2.70 V or above	2.5			—
				1.68 V or above (VCC) 1.65 V or above (VCC2)	2.5			—
TI SSP SS input hold time	Slave	3.00 V or above	$t_{TISH}$	2.5	—	ns		
				2.70 V or above	2.5			—
				1.68 V or above (VCC) 1.65 V or above (VCC2)	5.5			—
TI SSP next-access time	Slave	3.00 V or above	$t_{TIND}$	$2 \times t_{TCyc} + SLNDL \times t_{TCyc}$	—	ns		
				2.70 V or above	$2 \times t_{TCyc} + SLNDL \times t_{TCyc}$			—
				1.68 V or above (VCC) 1.65 V or above (VCC2)	$2 \times t_{TCyc} + SLNDL \times t_{TCyc}$			—
TI SSP master SS output delay	Master	3.00 V or above	$t_{TISSOD}$	—	4.0	ns	Figure 52.45	
				2.70 V or above	—			8.0
				1.68 V or above (VCC) 1.65 V or above (VCC2)	—			8.0
Data output delay time	Master	3.00 V or above	$t_{OD1}$	—	2.0	ns	Figure 52.43, Figure 52.44	
				2.70 V or above	—			3.0
				1.68 V or above (VCC) 1.65 V or above (VCC2)	—			6.0
		3.00 V or above	$t_{OD2}$	—	2.5			
				2.70 V or above	—			2.5
				1.68 V or above (VCC) 1.65 V or above (VCC2)	—			4.5
	Slave	3.00 V or above	$t_{OD}$	—	12.5			
				2.70 V or above	—			16.0
				1.68 V or above (VCC) 1.65 V or above (VCC2)	—			24.0
Data output hold time	Master	3.00 V or above	$t_{OH}$	-2.5	—	ns	Figure 52.43, Figure 52.44	
				2.70 V or above	-2.5			—
				1.68 V or above (VCC) 1.65 V or above (VCC2)	-4.5			—
	Slave	3.00 V or above	0.0	—				
			2.70 V or above	0.0	—			
			1.68 V or above (VCC) 1.65 V or above (VCC2)	0.0	—			

**Table 52.43 SPI timing (4 of 4)**

Conditions:

1. High drive output is selected in the Port Drive Capability bit in the PmnPFS register.
2. Use pins that have a letter appended to their names, for instance “\_A” or “\_B” to indicate group membership.
3. Load capacitance C = 15pF is applied to the VCC/VCC2 condition "3.00 V or above".

Parameter		VCC/VCC2	Symbol	Min	Max	Unit	Note
Successive transmission delay time	Master	3.00 V or above	$t_{TD}$	$t_{SPcyc} + 2 \times t_{TCyc}$	$8 \times t_{SPcyc} + 2 \times t_{TCyc}$	ns	Figure 52.43, Figure 52.44
		2.70 V or above		$t_{SPcyc} + 2 \times t_{TCyc}$	$8 \times t_{SPcyc} + 2 \times t_{TCyc}$		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		$t_{SPcyc} + 2 \times t_{TCyc}$	$8 \times t_{SPcyc} + 2 \times t_{TCyc}$		
	Slave	3.00 V or above		$t_{TCyc}$	—	ns	
		2.70 V or above		$t_{TCyc}$	—		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		$t_{TCyc}$	—		
MOSI and MISO rise and fall time	Output	3.00 V or above	$t_{Dr}, t_{Df}$	—	1.66	ns	Figure 52.43, Figure 52.44
		2.70 V or above		—	3.30		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.60		
	Input	3.00 V or above		—	1.0	$\mu s$	
		2.70 V or above		—	1.0		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	1.0		
SSL rise and fall time	Output	3.00- V or above	$t_{SSLr}, t_{SSLf}$	—	1.66	ns	Figure 52.43, Figure 52.44
		2.70 V or above		—	3.30		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	6.60		
	Input	3.00 V or above		—	1.0	$\mu s$	
		2.70 V or above		—	1.0		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	1.0		
Slave access time	Slave	3.00 V or above	$t_{SA}$	—	20.0	ns	Figure 52.46, Figure 52.47
		2.70 V or above		—	20.0		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	25.0		
Slave output release time	Slave	3.00 V or above	$t_{REL}$	—	20.0	ns	
		2.70 V or above		—	20.0		
		1.68 V or above (VCC) 1.65 V or above (VCC2)		—	25.0		

Note 1. 1  $\mu s$  at the longest

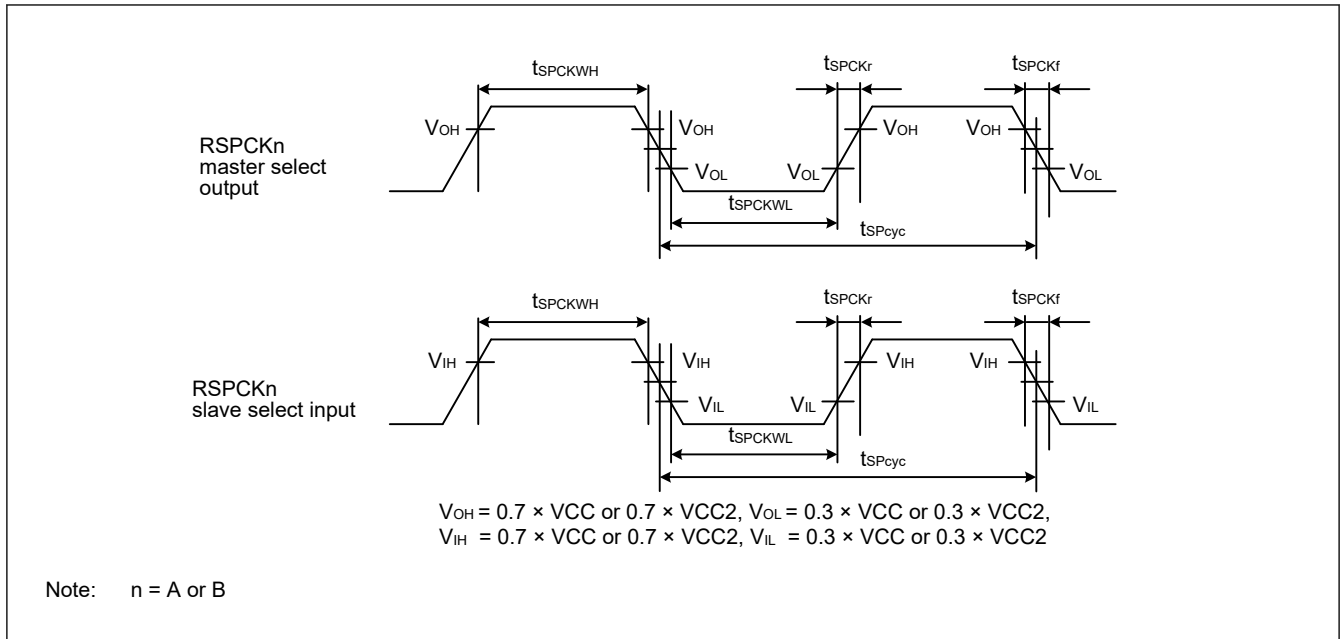


Figure 52.42 SPI clock timing

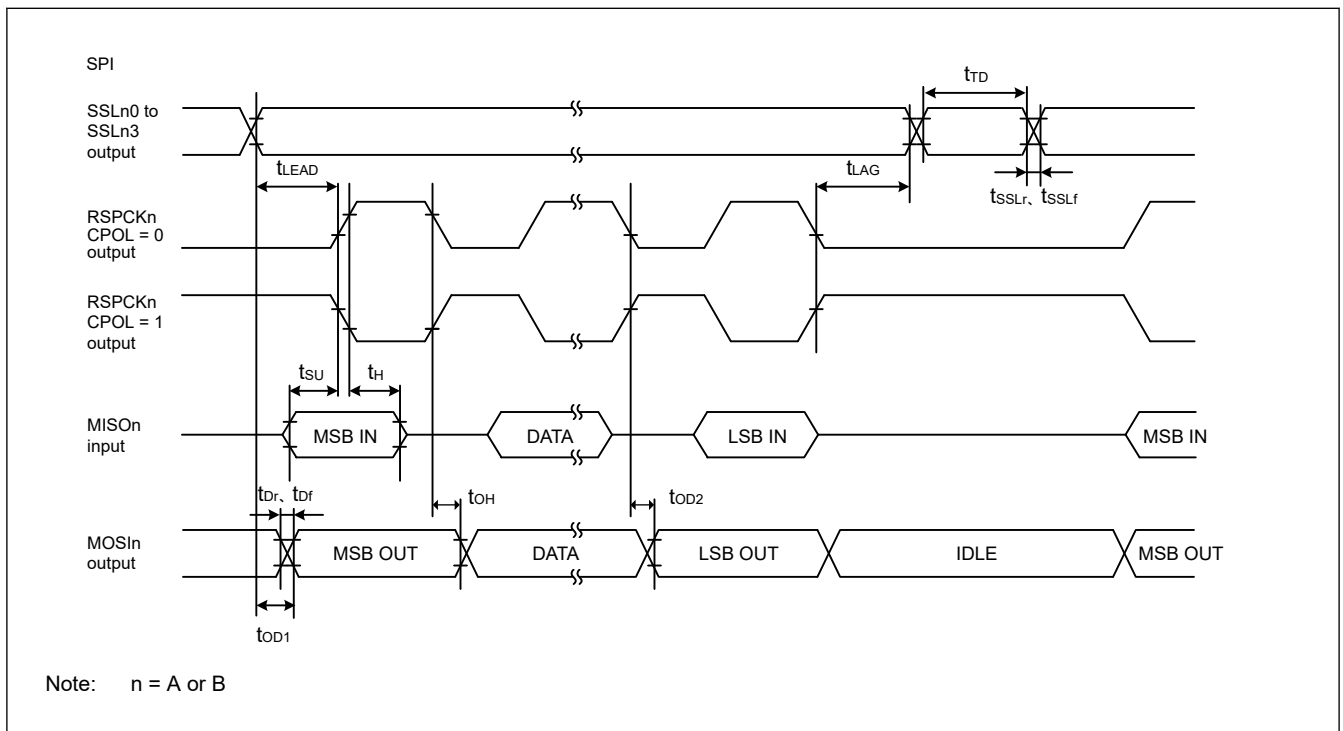


Figure 52.43 SPI timing for Motorola SPI master when CPHA = 0

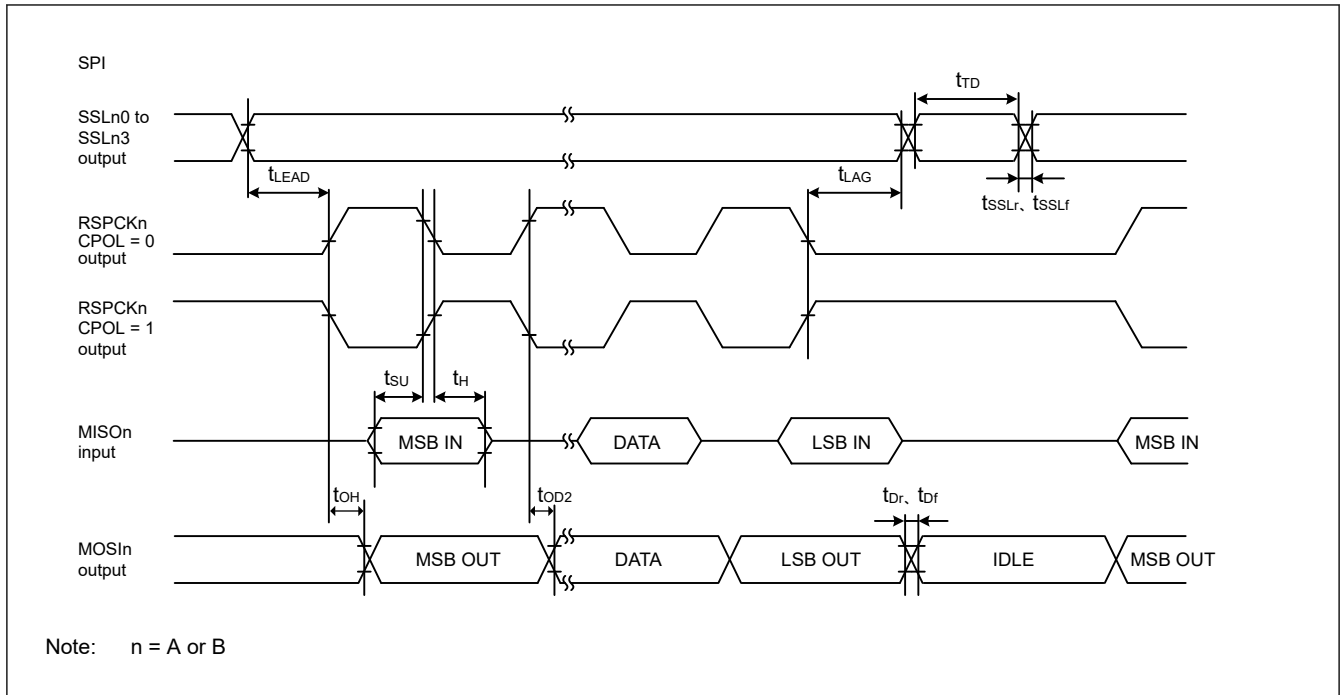


Figure 52.44 SPI timing for Motorola SPI master when CPHA = 1

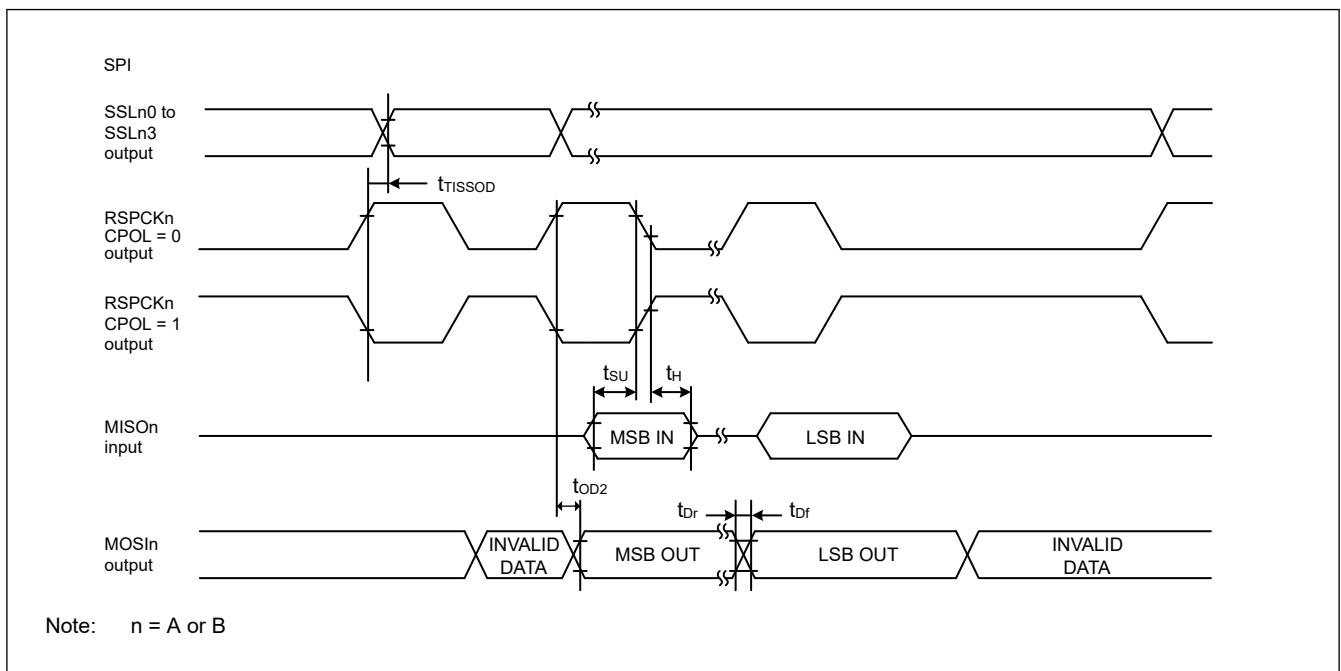


Figure 52.45 SPI timing for TI SSP master

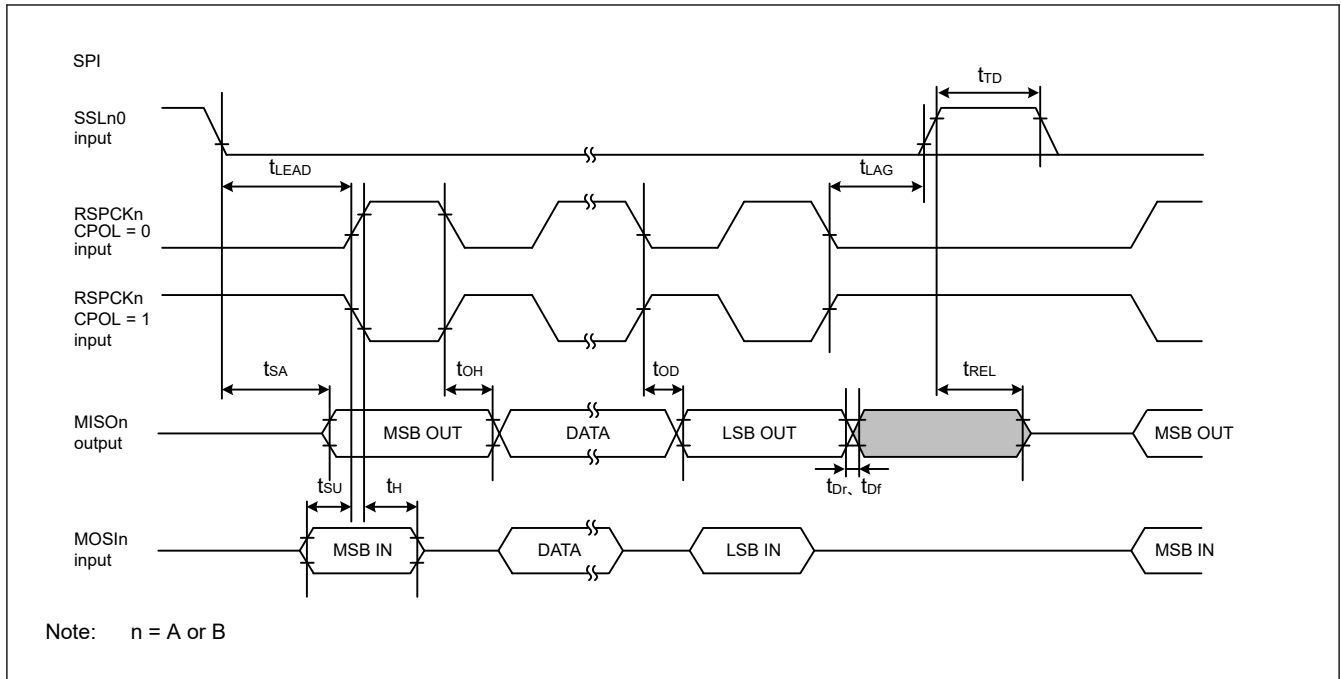


Figure 52.46 SPI timing for Motorola SPI slave when CPHA = 0

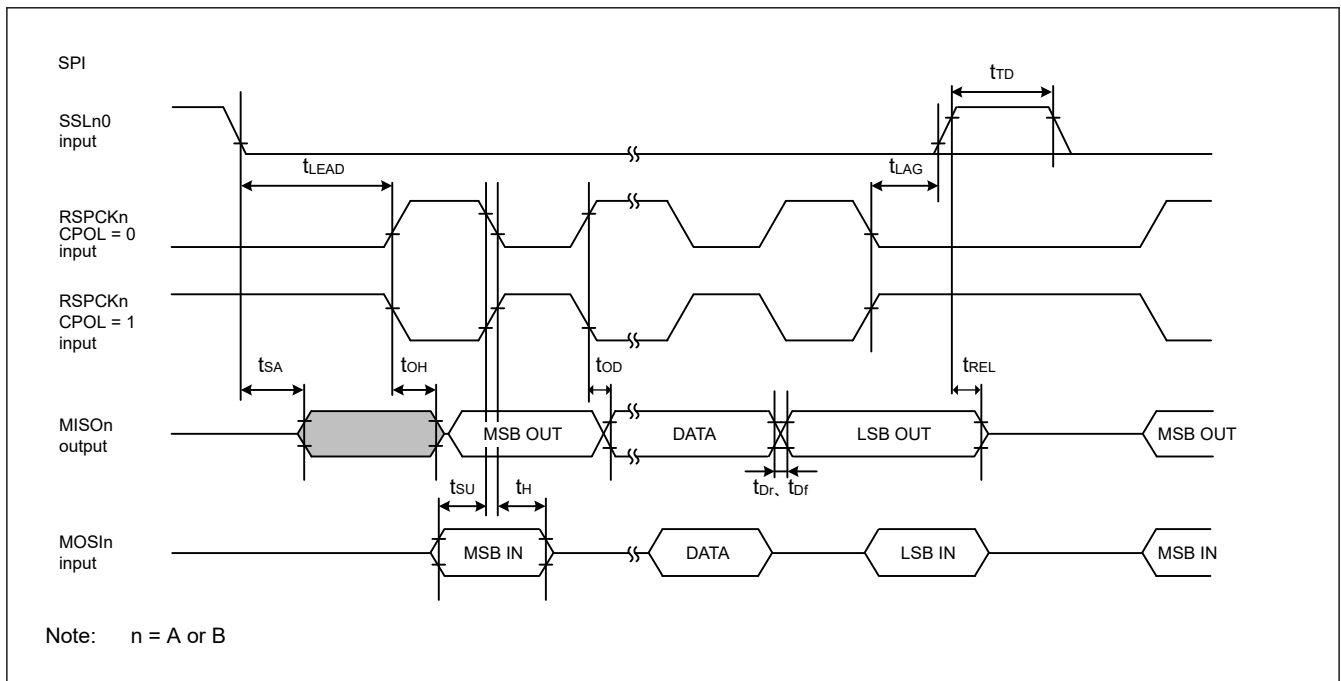


Figure 52.47 SPI timing for Motorola SPI slave when CPHA = 1

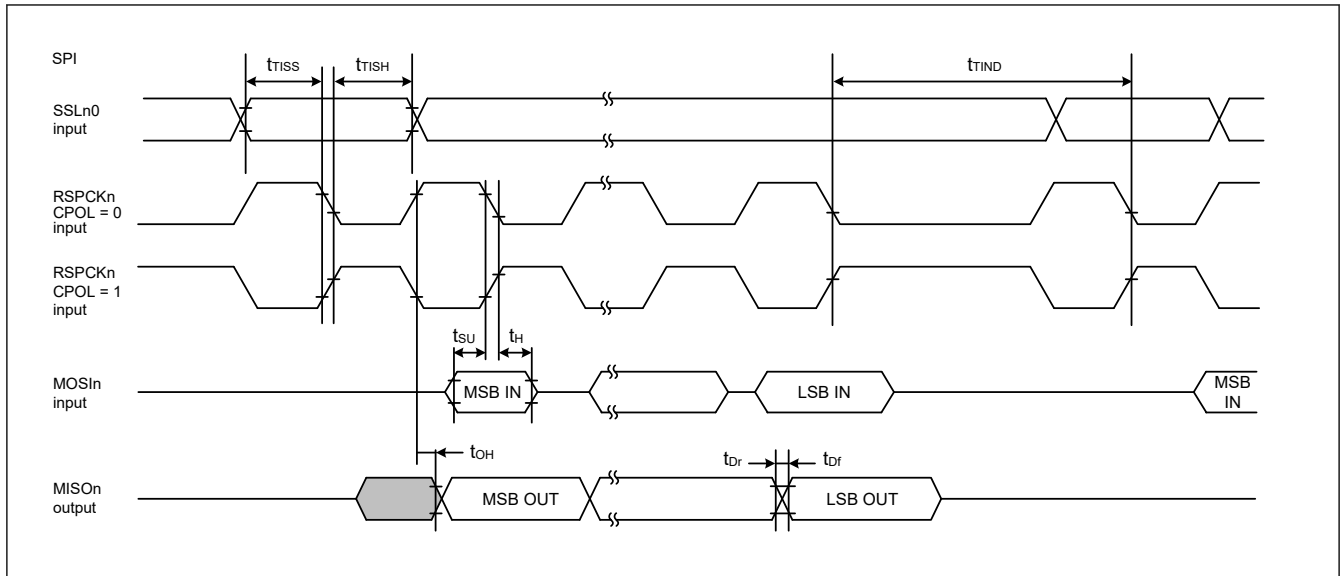


Figure 52.48 SPI timing for TI SSP slave when transmit with delay between frames

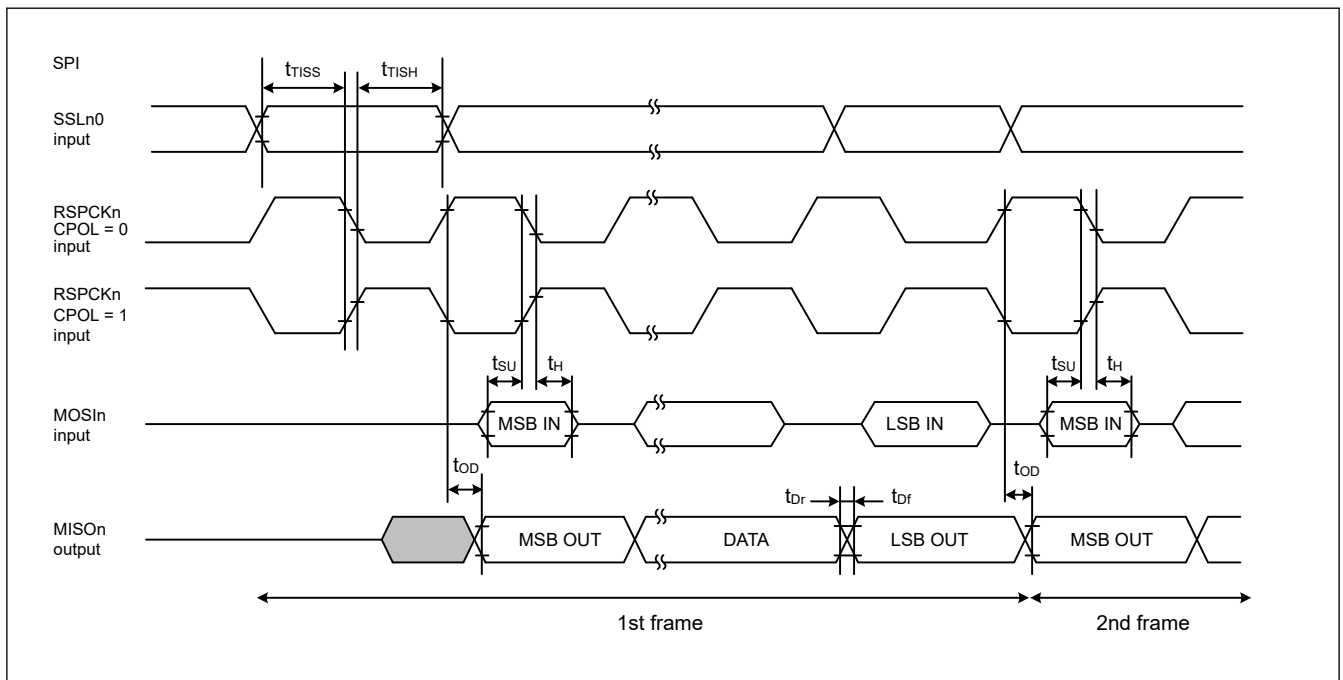


Figure 52.49 SPI timing for TI SSP slave when transmit with no delay between frames



## 52.3.10 OSPI Timing

**Table 52.44 OSPI timing (1 of 2)**

Conditions:

High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM\_SCLK, OM\_SCLKN, OM\_SIO7-0, OM\_DQS.

High drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM\_CS0, OM\_CS1.

Load capacitance C = 20 pF

Item	Symbol	VCC/VCC2	Min	Max	Unit	Note
Cycle time	SDR without OM_DQS	2.70 V or above	20	—	ns	Figure 52.50
		1.65 V to 2.00 V	20	—		
	SDR with OM_DQS/DDR	2.70 V or above	10	—	ns	
		1.65 V to 2.00 V	10	—		
Clock output slew rate	$t_{SRck}$	2.70 V or above	0.56	—	V/ns	
		1.65 V to 2.00 V	0.56	—		
Clock Duty cycle-distortion	$t_{CKDCD}$	2.70 V or above	0	0.5	ns	
		1.65 V to 2.00 V	0	0.5		
Clock Minimum Pulse width	$t_{CKMPW}$	2.70 V or above	4.5	—	ns	
		1.65 V to 2.00 V	4.5	—		
Differential clock crossing volate	$V_{ox}(AC)$	2.70 V or above	$0.4 \times VCC2$	$0.6 \times VCC2$	V	
		1.65 V to 2.00 V	$0.4 \times VCC2$	$0.6 \times VCC2$		
DS Duty cycle distortion	$t_{DSDCD}$	2.70 V or above	0	0.4	ns	
		1.65 V to 2.00 V	0	0.4		
DS Minimum Pulse width	$t_{DSMPW}$	2.70 V or above	4.1	—	ns	
		1.65 V to 2.00 V	4.1	—		
Data input/output slew rate	$t_{SR}$	2.70 V or above	1.03	—	ns	
		1.65 V to 2.00 V	0.56	—		
Data input setup time (to OM_SCLK/ OM_SCLKN)	$t_{SU}$	2.70 V or above	8.0	—	ns	Figure 52.51
		1.65 V to 2.00 V	12.5	—		
Data input hold time (to OM_SCLK/ OM_SCLKN)	$t_H$	2.70 V or above	0.5	—	ns	
		1.65 V to 2.00 V	0.5	—		
Data output valid time	$t_{OV}^{*2}$	2.70 V or above	—	4.0	ns	
		1.65 V to 2.00 V	—	4.0		
Data output hold time	$t_{OH}$	2.70 V or above	-2.0	—	ns	
		1.65 V to 2.00 V	-2.0	—		
Data output buffer off time	$t_{BOFF}$	2.70 V or above	-2.0	—	ns	
		1.65 V to 2.00 V	-2.0	—		

**Table 52.44 OSPI timing (2 of 2)**

Conditions:

High-speed high drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM\_SCLK, OM\_SCLKN, OM\_SIO7-0, OM\_DQS.

High drive output is selected in the port drive capability bit in the PmnPFS register for the following pins: OM\_CS0, OM\_CS1.

Load capacitance C = 20 pF

Item	Symbol	VCC/VCC2	Min	Max	Unit	Note	
Data input setup time (to OM_DQS)	$t_{SU}$	2.70 V or above	-0.9	—	ns	Figure 52.52, Figure 52.53	
		1.65 V to 2.00 V	-0.9	—			
Data input hold time (to OM_DQS)	$t_H$	2.70 V or above	3.2	—	ns		
		1.65 V to 2.00 V	3.2	—			
Data output valid time	$t_{OV}^{*2}$	2.70 V or above	—	$t_{PERIOD}/4 + 0.9$	ns		
		1.65 V to 2.00 V	—	$t_{PERIOD}/4 + 0.9$			
Data output hold time	$t_{HO}$	2.70 V or above	1.1	—	ns		
		1.65 V to 2.00 V	1.1	—			
Data output buffer off time	$t_{BOFF}$	2.70 V or above	1.1	—	ns		
		1.65 V to 2.00 V	1.1	—			
Clock Low to CS Low	$t_{CKLCSL}$	2.70 V or above	8	—	ns	Figure 52.51, Figure 52.52, Figure 52.53	
		1.65 V to 2.00 V	8	—			
CS Low to Clock High	$t_{CSLCKH}^{*3}$	2.70 V or above	8	—	ns		
		1.65 V to 2.00 V	8	—			
Clock Low to CS High	$t_{CKLCSH}$	2.70 V or above	8	—	ns		
		1.65 V to 2.00 V	8	—			
CS High to Clock High	$t_{CSCHKH}$	2.70 V or above	8	—	ns		
		1.65 V to 2.00 V	8	—			
DS Low output to CS High	$t_{DSLCSH}$	2.70 V or above	$0.8 \times t_{PERIOD}$	—	ns		Figure 52.54
		1.65 V to 2.00 V	$0.8 \times t_{PERIOD}$	—			
CS High to DS Tri-State	$t_{CSHDST}$	2.70 V or above	—	$t_{PERIOD}$	ns		
		1.65 V to 2.00V	—	$t_{PERIOD}$			
CS Low to DS Low input <sup>*1</sup>	$t_{CSLDSL}$	2.70 V or above	0	18.5	ns		
		1.65 V to 2.00 V	0	12.5			
DS Tri-State to CS Low	$t_{DSTCSL}$	2.70 V or above	0	—	ns		
		1.65 V to 2.00 V	0	—			

Note 1. This restriction does not need to be met when using the JESD251 Profile 1.0 memory with an external pull-down attached to the OM\_DQS pin.

Note 2. Condition: COMCFG.OEASTEX = 1

Note 3. Condition: LIOCFGCSx.CSASTEX = 1

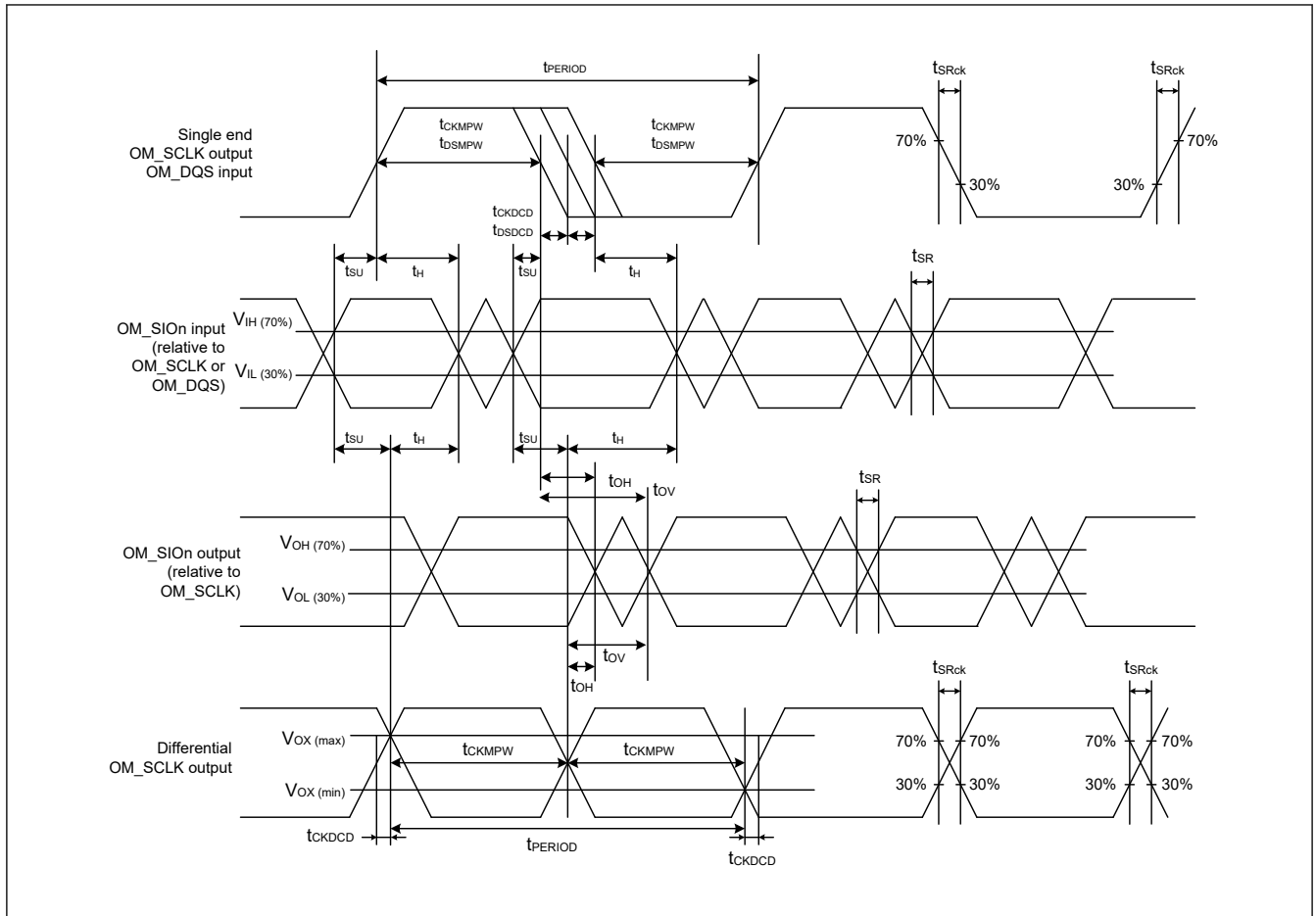


Figure 52.50 OSPI clock / DS timing

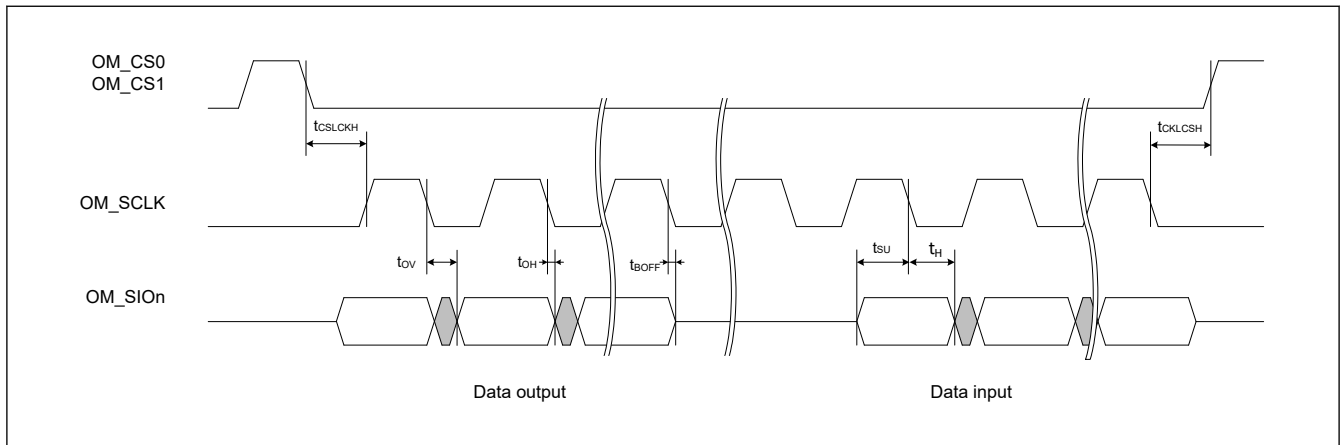


Figure 52.51 SDR transmit/receive timing (1S-1S-1S, 1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S)

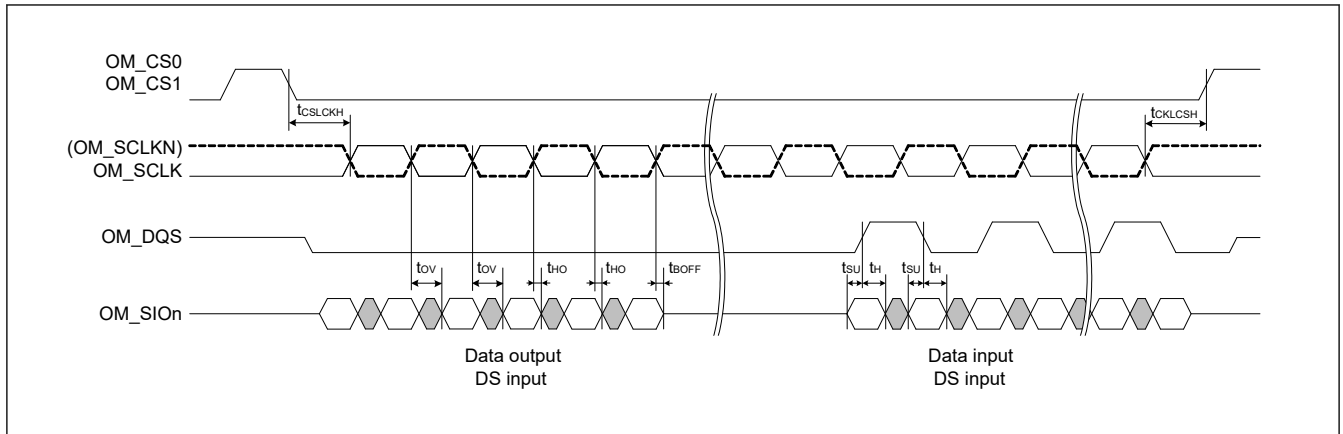


Figure 52.52 DDR transmit/receive timing (4S-4D-4D, 8D-8D-8D)

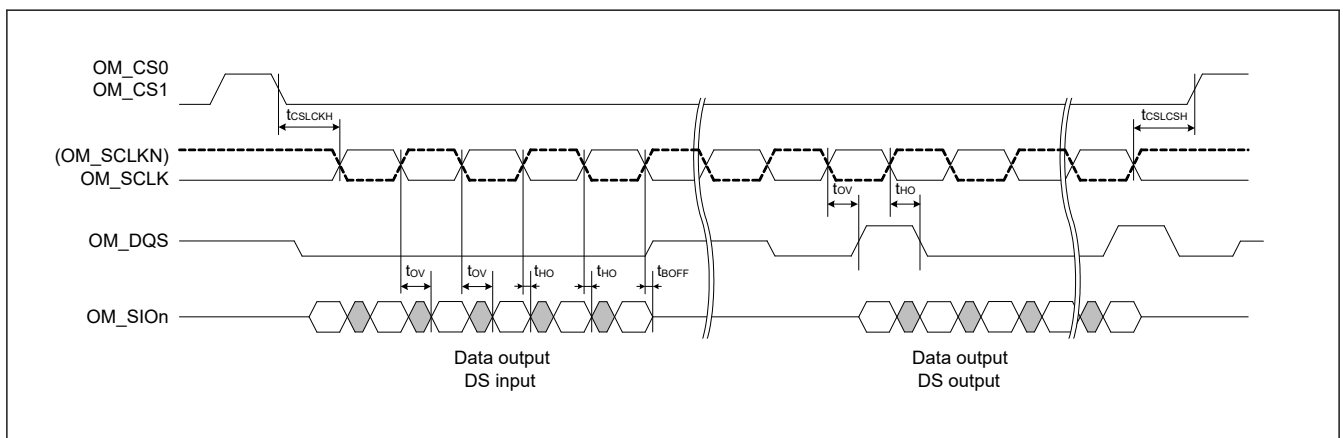


Figure 52.53 DDR transmit/receive timing (HyperRAM write)

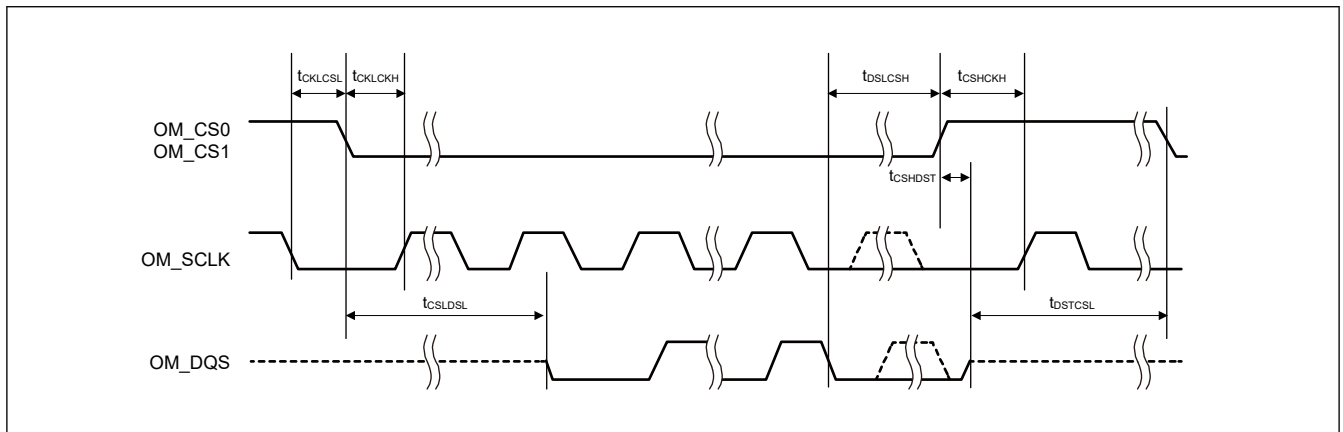


Figure 52.54 DS to CS signal timing

## 52.3.11 IIC Timing

**Table 52.45 IIC timing (1) (1 of 2)**

(1) Conditions: Middle drive output is selected when VCC is 2.70 V or above, High drive output is selected when VCC is 1.68 to 1.95 V in the port drive capability bit in the PmnPFS register for the following pins: SDA0\_B, SCL0\_B, SDA1\_B, SCL1\_B

(2) The following pins do not require setting: SCL0\_A, SDA0\_A, SCL1\_A, SDA1\_A

(3) Use pins that have a letter appended to their names, for instance “\_A” or “\_B”, to indicate group membership.

For the IIC interface, the AC portion of the electrical characteristics is measured for each group.

Parameter	Symbol	VCC	Min	Max	Unit	Test conditions	
IIC (Standard mode, SMBus) ICFER.FMPE = 0 when VCC is 2.70 V or above, ICFER.FMPE = 1 when VCC is 1.68 to 1.95 V	SCL input cycle time	$t_{SCL}$	2.70 V or above 1.68 to 1.95 V	$6 (12) \times t_{IICcyc} + 1300$	—	ns	Figure 52.55
	SCL input high pulse width	$t_{SCLH}$	2.70 V or above 1.68 to 1.95 V	$3 (6) \times t_{IICcyc} + 300$	—	ns	
	SCL input low pulse width	$t_{SCLL}$	2.70 V or above 1.68 to 1.95 V	$3 (6) \times t_{IICcyc} + 300$	—	ns	
	SCL, SDA rise time	$t_{Sr}$	2.70 V or above 1.68 to 1.95 V	—	1000	ns	
	SCL, SDA fall time	$t_{Sf}$	2.70 V or above 1.68 to 1.95 V	—	300	ns	
	SCL, SDA input spike pulse removal time	$t_{SP}$	2.70 V or above 1.68 to 1.95 V	0	$1 (4) \times t_{IICcyc}$	ns	
	SDA input bus free time when wakeup function is disabled	$t_{BUF}$	2.70 V or above 1.68 to 1.95 V	$3 (6) \times t_{IICcyc} + 300$	—	ns	
	SDA input bus free time when wakeup function is enabled	$t_{BUF}$	2.70 V or above 1.68 to 1.95 V	$3 (6) \times t_{IICcyc} + 4 \times t_{Pcyc} + 300$	—	ns	
	START condition input hold time when wakeup function is disabled	$t_{STAH}$	2.70 V or above 1.68 to 1.95 V	$t_{IICcyc} + 300$	—	ns	
	START condition input hold time when wakeup function is enabled	$t_{STAH}$	2.70 V or above 1.68 to 1.95 V	$1 (5) \times t_{IICcyc} + t_{Pcyc} + 300$	—	ns	
	Repeated START condition input setup time	$t_{STAS}$	2.70 V or above 1.68 to 1.95 V	1000	—	ns	
	STOP condition input setup time	$t_{STOS}$	2.70 V or above 1.68 to 1.95 V	1000	—	ns	
	Data input setup time	$t_{SDAS}$	2.70 V or above 1.68 to 1.95 V	$t_{IICcyc} + 50$	—	ns	
	Data input hold time	$t_{SDAH}$	2.70 V or above 1.68 to 1.95 V	0	—	ns	
	SCL, SDA capacitive load	$C_b$	2.70 V or above 1.68 to 1.95 V	—	400	pF	

**Table 52.45 IIC timing (1) (2 of 2)**

(1) Conditions: Middle drive output is selected when VCC is 2.70 V or above, High drive output is selected when VCC is 1.68 to 1.95 V in the port drive capability bit in the PmnPFS register for the following pins: SDA0\_B, SCL0\_B, SDA1\_B, SCL1\_B

(2) The following pins do not require setting: SCL0\_A, SDA0\_A, SCL1\_A, SDA1\_A

(3) Use pins that have a letter appended to their names, for instance "\_A" or "\_B", to indicate group membership.

For the IIC interface, the AC portion of the electrical characteristics is measured for each group.

Parameter	Symbol	VCC	Min	Max	Unit	Test conditions
IIC (Fast mode) ICFER.FMPE = 0 when VCC is 2.70 V or above, ICFER.FMPE = 1 when VCC is 1.68 to 1.95 V	SCL input cycle time	2.70 V or above	6 (12) × t <sub>IICcyc</sub> + 600	—	ns	Figure 52.55
		1.68 to 1.95 V				
	SCL input high pulse width	2.70 V or above	3 (6) × t <sub>IICcyc</sub> + 300	—	ns	
		1.68 to 1.95 V				
	SCL input low pulse width	2.70 V or above	3 (6) × t <sub>IICcyc</sub> + 300	—	ns	
		1.68 to 1.95 V				
	SCL, SDA rise time	2.70 V or above	20	300	ns	
		1.68 to 1.95 V				
	SCL, SDA fall time	2.70 V or above	20 × (external pullup voltage/5.5 V) <sup>*1</sup>	300	ns	
		1.68 to 1.95 V				
	SCL, SDA input spike pulse removal time	2.70 V or above	0	1 (4) × t <sub>IICcyc</sub>	ns	
		1.68 to 1.95 V				
	SDA input bus free time when wakeup function is disabled	2.70 V or above	3 (6) × t <sub>IICcyc</sub> + 300	—	ns	
		1.68 to 1.95 V				
	SDA input bus free time when wakeup function is enabled	2.70 V or above	3 (6) × t <sub>IICcyc</sub> + 4 × t <sub>Pcyc</sub> + 300	—	ns	
		1.68 to 1.95 V				
START condition input hold time when wakeup function is disabled	2.70 V or above	t <sub>IICcyc</sub> + 300	—	ns		
	1.68 to 1.95 V					
START condition input hold time when wakeup function is enabled	2.70 V or above	1 (5) × t <sub>IICcyc</sub> + t <sub>Pcyc</sub> + 300	—	ns		
	1.68 to 1.95 V					
Repeated START condition input setup time	2.70 V or above	300	—	ns		
	1.68 to 1.95 V					
STOP condition input setup time	2.70 V or above	300	—	ns		
	1.68 to 1.95 V					
Data input setup time	2.70 V or above	t <sub>IICcyc</sub> + 50	—	ns		
	1.68 to 1.95 V					
Data input hold time	2.70 V or above	0	—	ns		
	1.68 to 1.95 V					
SCL, SDA capacitive load	2.70 V or above	—	400	pF		
	1.68 to 1.95 V					

Note: t<sub>IICcyc</sub>: IIC internal reference clock (IICφ) cycle, t<sub>Pcyc</sub>: PCLKB cycle.

Note: Values in parentheses apply when ICMR3.NF[1:0] is set to 11b while the digital filter is enabled with ICFER.NFE set to 1.

Note: Must use pins that have a letter appended to their name, for instance "\_A", "\_B", to indicate group membership. For the IIC interface, the AC portion of the electrical characteristics is measured for each group.

Note 1. Only supported for SCL0\_A, SDA0\_A, SCL1\_A, and SDA1\_A.

**Table 52.46 IIC timing (2)**

Setting of the SCL0\_A, SDA0\_A, SCL1\_A, SDA1\_A pins are not required with the port drive capability bit in the PmnPFS register.

Parameter		Symbol	VCC	Min	Max	Unit	Test conditions
IIC (Fast-mode+) ICFER.FMPE = 1	SCL input cycle time	$t_{SCL}$	2.70 V or above	$6 (12) \times t_{IICcyc} + 240$	—	ns	Figure 52.55
	SCL input high pulse width	$t_{SCLH}$	2.70 V or above	$3 (6) \times t_{IICcyc} + 120$	—	ns	
	SCL input low pulse width	$t_{SCLL}$	2.70 V or above	$3 (6) \times t_{IICcyc} + 120$	—	ns	
	SCL, SDA rise time	$t_{Sr}$	2.70 V or above	—	120	ns	
	SCL, SDA fall time	$t_{Sf}$	2.70 V or above	$20 \times (\text{external pullup voltage} / 5.5V)$	120	ns	
	SCL, SDA input spike pulse removal time	$t_{SP}$	2.70 V or above	0	$1 (4) \times t_{IICcyc}$	ns	
	SDA input bus free time when wakeup function is disabled	$t_{BUF}$	2.70 V or above	$3 (6) \times t_{IICcyc} + 120$	—	ns	
	SDA input bus free time when wakeup function is enabled	$t_{BUF}$	2.70 V or above	$3 (6) \times t_{IICcyc} + 4 \times t_{Pcyc} + 120$	—	ns	
	Start condition input hold time when wakeup function is disabled	$t_{STAH}$	2.70 V or above	$t_{IICcyc} + 120$	—	ns	
	START condition input hold time when wakeup function is enabled	$t_{STAH}$	2.70 V or above	$1 (5) \times t_{IICcyc} + t_{Pcyc} + 120$	—	ns	
	Restart condition input setup time	$t_{STAS}$	2.70 V or above	120	—	ns	
	Stop condition input setup time	$t_{STOS}$	2.70 V or above	120	—	ns	
	Data input setup time	$t_{SDAS}$	2.70 V or above	$t_{IICcyc} + 30$	—	ns	
	Data input hold time	$t_{SDAH}$	2.70 V or above	0	—	ns	
SCL, SDA capacitive load	$C_b^{*1}$	2.70 V or above	—	550	pF		

Note:  $t_{IICcyc}$ : IIC internal reference clock (IIC $\phi$ ) cycle,  $t_{Pcyc}$ : PCLKB cycle.

Note: Values in parentheses apply when ICMR3.NF[1:0] is set to 11b while the digital filter is enabled with ICFER.NFE set to 1.

Note 1.  $C_b$  indicates the total capacity of the bus line.

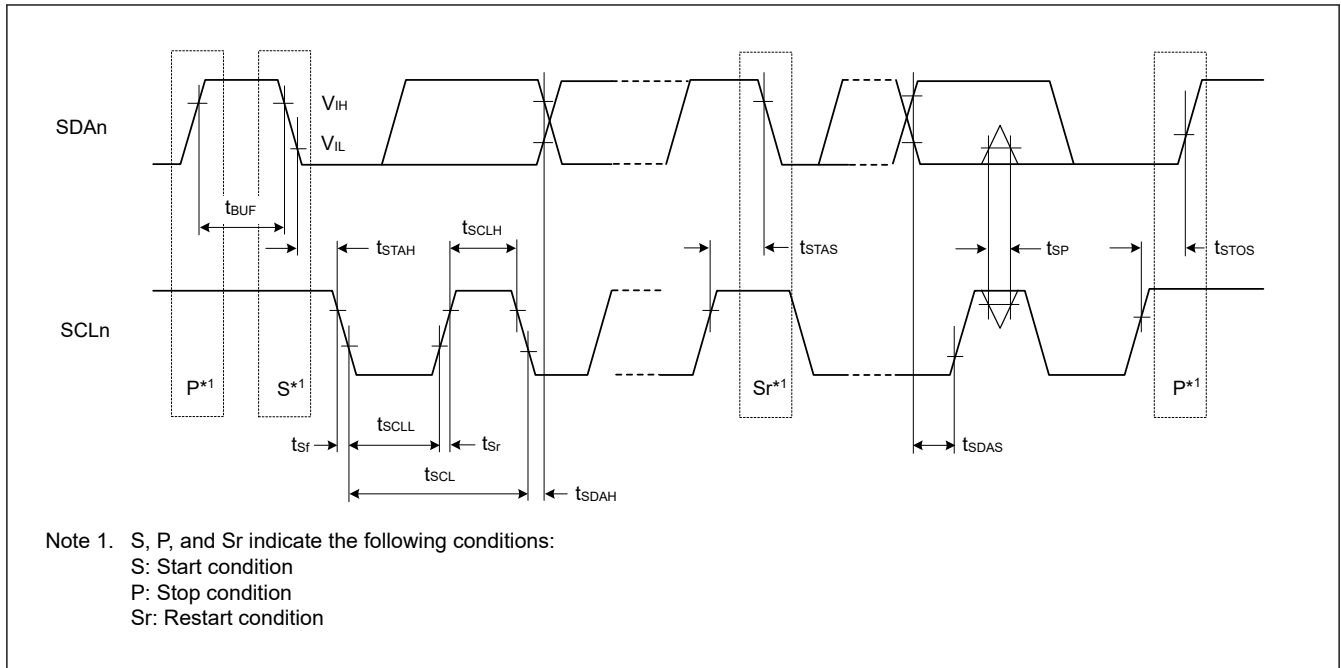


Figure 52.55 I<sup>2</sup>C bus interface input/output timing

52.3.12 SSIE Timing

Table 52.47 SSIE timing (1 of 2)

(1) High drive output is selected with the Port Drive Capability bit in the PmnPFS register.  
 (2) Use pins that have a letter appended to their names, for instance “\_A”, “\_B” or “\_C” to indicate group membership. For the SSIE interface, the AC portion of the electrical characteristics is measured for each group.

Parameter			Symbol	VCC	Min.	Max.	Unit	Comments
SSIBCK	Cycle	Master	$t_O$	2.70V or above	80	—	ns	Figure 52.56
				1.68V or above	80	—		
		Slave	$t_I$	2.70V or above	80	—	ns	
				1.68V or above	80	—		
	High level/low level	Master	$t_{HC}/t_{LC}$	2.70V or above	0.35	—	$t_O$	
				1.68V or above	0.35	—		
		Slave		2.70V or above	0.35	—	$t_I$	
				1.68V or above	0.35	—		
Rising time/falling time	Master	$t_{RC}/t_{FC}$	2.70V or above	—	0.15	$t_O / t_I$		
			1.68V or above	—	0.15			
	Slave		2.70V or above	—	0.15	$t_O / t_I$		
			1.68V or above	—	0.15			



**Table 52.47 SSIE timing (2 of 2)**

(1) High drive output is selected with the Port Drive Capability bit in the PmnPFS register.

(2) Use pins that have a letter appended to their names, for instance “\_A”, “\_B” or “\_C” to indicate group membership. For the SSIE interface, the AC portion of the electrical characteristics is measured for each group.

Parameter			Symbol	VCC	Min.	Max.	Unit	Comments
SSILRCK0/ SSIFS0, SSITXD0, SSIRXD0	Input set up time	Master	$t_{SR}$	2.70V or above	12	—	ns	Figure 52.58, Figure 52.59
				1.68V or above	20	—		
		Slave		2.70V or above	12	—	ns	
				1.68V or above	12	—		
	Input hold time	Master	$t_{HR}$	2.70V or above	8	—	ns	
				1.68V or above	8	—		
		Slave		2.70V or above	15	—	ns	
				1.68V or above	15	—		
	Output delay time	Master	$t_{DTR}$	2.70V or above	-10	5	ns	
				1.68V or above	-10	7		
		Slave		2.70V or above	0	20	ns	
				1.68V or above	0	25		
Output delay time from SSILRCK/ SSIFS change	Slave	$t_{DTRW}$	2.70V or above	—	20	ns	Figure 52.60	
			1.68V or above	—	25			
GTIOC2A, AUDIO_CLK	Cycle	$t_{EXcyc}$	2.70V or above	20	—	ns	Figure 52.57*1	
			1.68V or above	40	—			
	High level/ low level	$t_{EXL}/t_{EXH}$	2.70V or above	0.4	—	$t_{EXcyc}$		
			1.68V or above	0.4	—			
	Rising time/ falling time	$t_{EXr}/t_{EXf}$	2.70V or above	—	0.1*2	$t_{EXcyc}$		
			1.68V or above	—	0.1*2			

Note 1. For slave-mode transmission, SSIE has a path, through which the signal input from the SSILRCK/SSIFS pin is used to generate transmit data, and the transmit data is logically output to the SSITXD0 or SSIDATA1 pin.

Note 2. 1 $\mu$ s at the longest

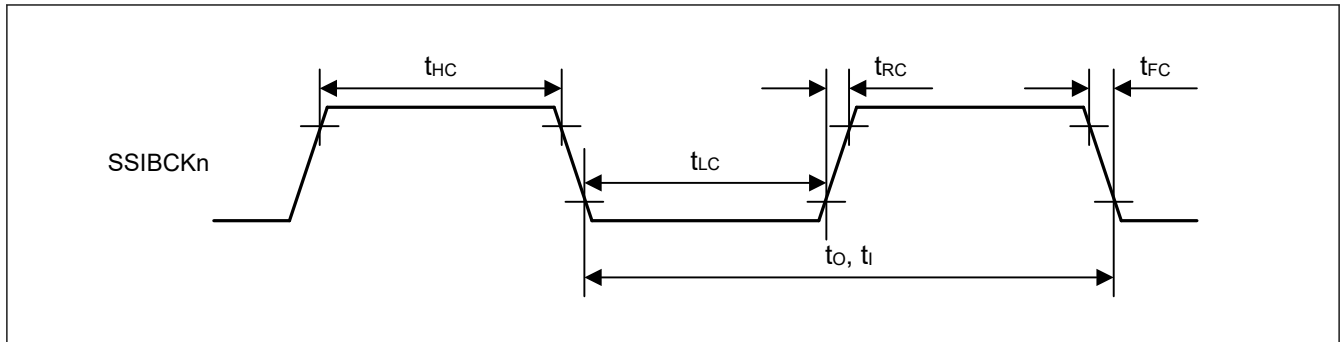


Figure 52.56 SSIE clock input/output timing

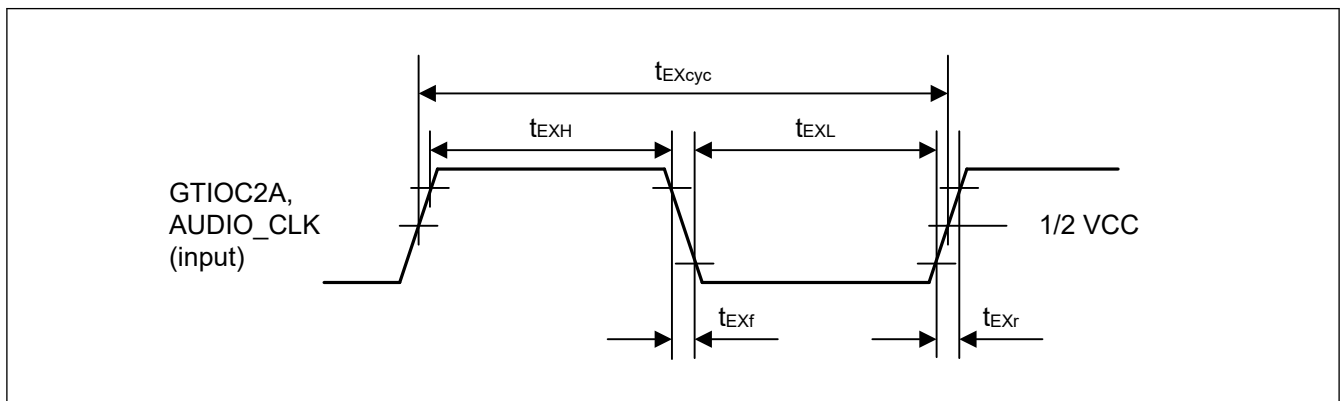


Figure 52.57 Clock input timing

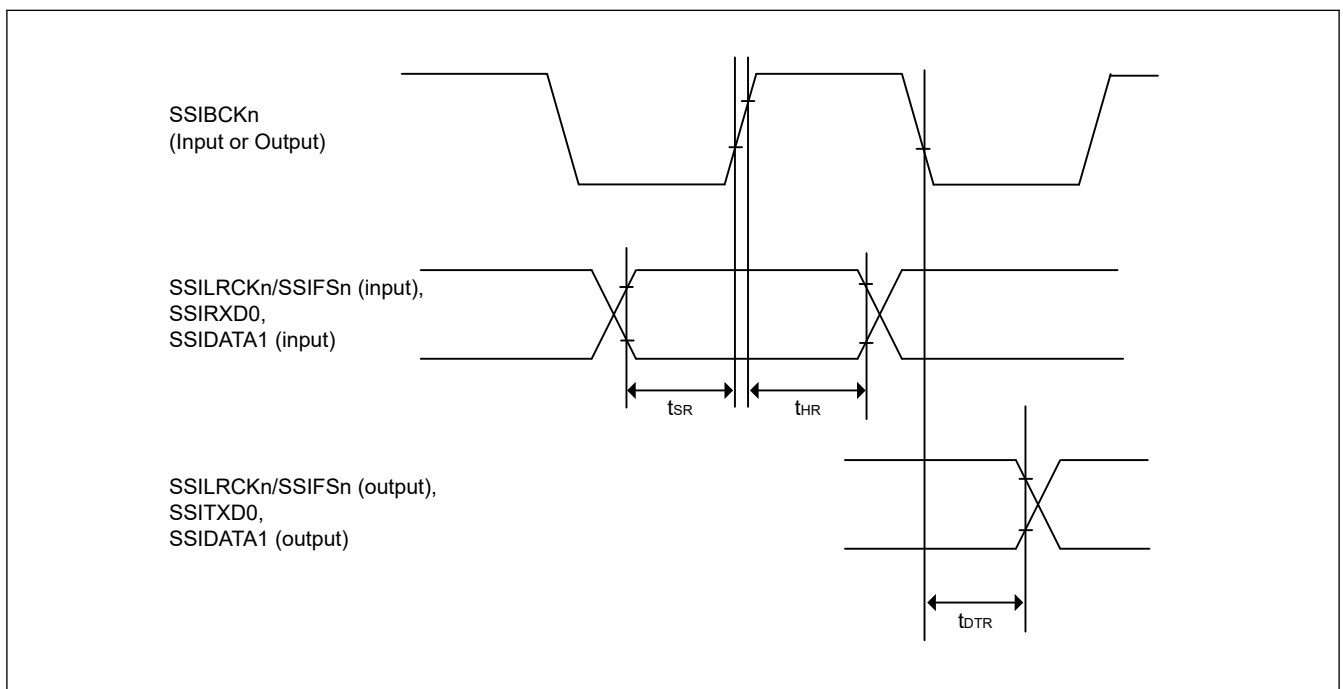


Figure 52.58 SSIE data transmit and receive timing when SSICR.BCKP = 0

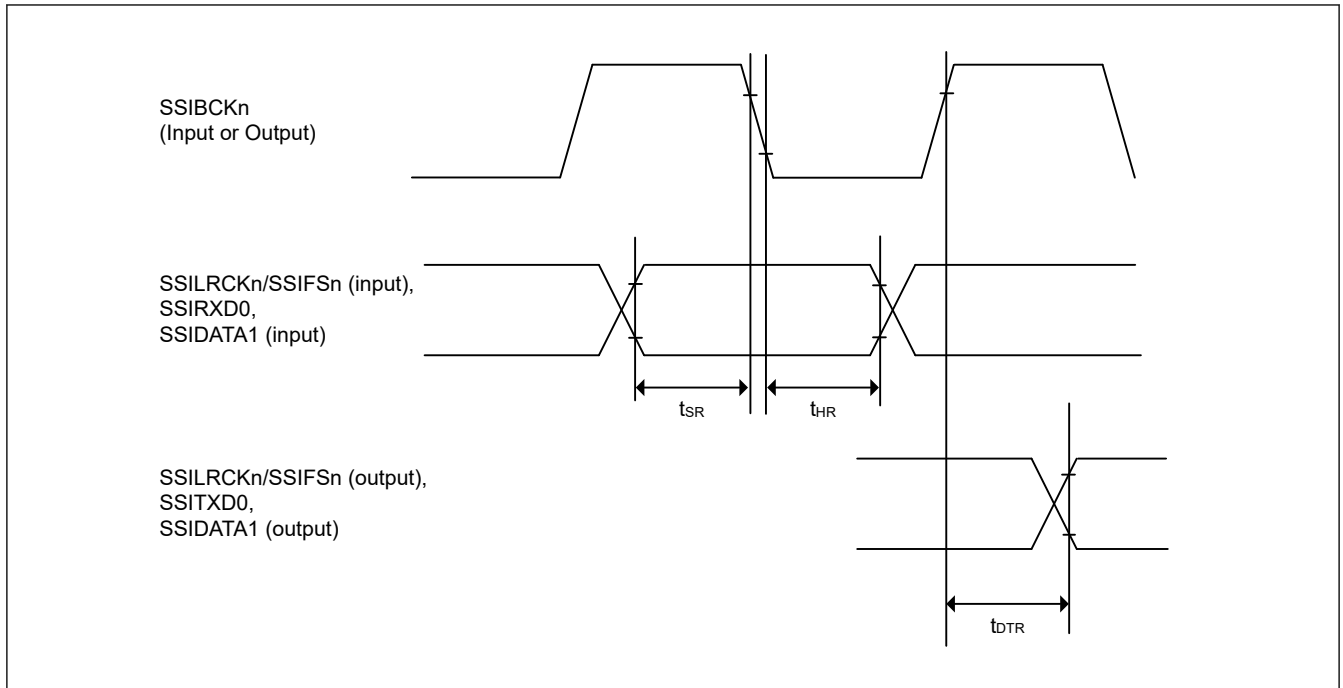


Figure 52.59 SSIE data transmit and receive timing when SSICR.BCKP = 1

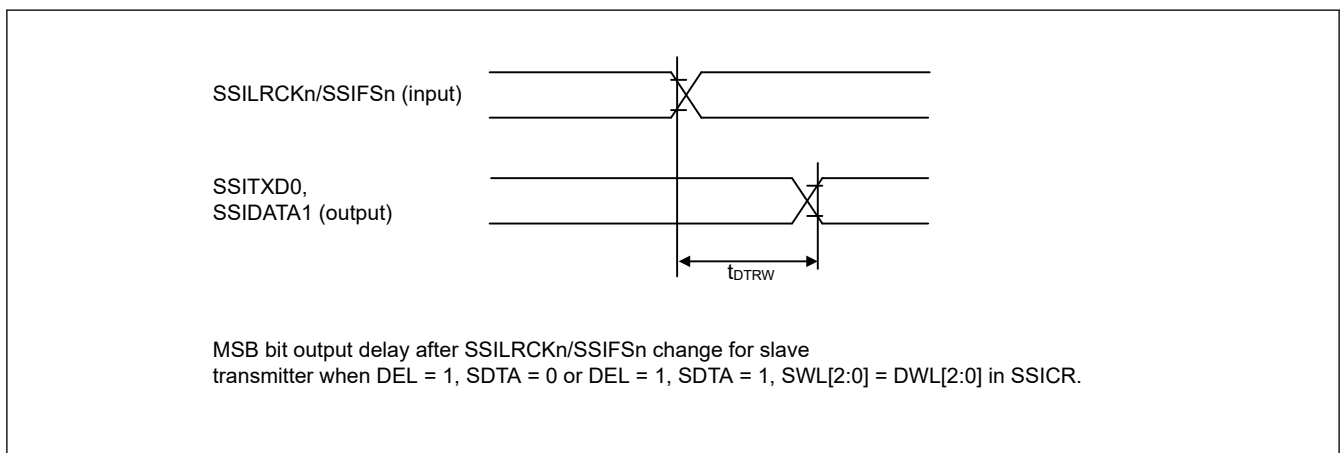


Figure 52.60 SSIE data output delay after SSILRCK0/SSIFSn change

## 52.3.13 ETHERC Timing

**Table 52.48 ETHERC timing**

Conditions: ETHERC (RMII): Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register for the following pins: ET0\_MDC, ET0\_MDIO.

For other pins, high drive output is selected in the Port Drive Capability bit in the PmnPFS register.

ETHERC (MII): Middle drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter	Symbol	VCC	Min	Max	Unit	Test conditions	
ETHERC (RMII)	REF50CK0 cycle time	$T_{ck}$	2.70V or above	20	—	ns	Figure 52.61 to Figure 52.64
	REF50CK0 frequency, typical 50 MHz	—	—	50 + 100 ppm	MHz		
	REF50CK0 duty	—	35	65	%		
	REF50CK0 rise/fall time	$T_{ckr/ckf}$	0.5	3.5	ns		
	RMII_XXXX <sup>*1</sup> output delay	$T_{co}$	2.5	12.0	ns		
	RMII_XXXX <sup>*2</sup> setup time	$T_{su}$	3	—	ns		
	RMII_XXXX <sup>*2</sup> hold time	$T_{hd}$	1	—	ns		
	RMII_XXXX <sup>*1, *2</sup> rise/fall time	$T_r/T_f$	0.5	5.0	ns		
	ET0_WOL output delay	$t_{WOLd}$	1	23.5	ns	Figure 52.65	
ETHERC (MII)	ET0_TX_CLK cycle time	$t_{Tcyc}$	40	—	ns	—	
	ET0_TX_EN output delay	$t_{TENd}$	1	20	ns	Figure 52.66	
	ET0_ETXD0 to ET_ETXD3 output delay	$t_{MTDd}$	1	20	ns	Figure 52.67	
	ET0_CRS setup time	$t_{CRSs}$	10	—	ns		
	ET0_CRS hold time	$t_{CRSh}$	10	—	ns		
	ET0_COL setup time	$t_{COLs}$	10	—	ns		
	ET0_COL hold time	$t_{COLh}$	10	—	ns	Figure 52.68	
	ET0_RX_CLK cycle time	$t_{TRcyc}$	40	—	ns		
	ET0_RX_DV setup time	$t_{RDVs}$	10	—	ns	Figure 52.69	
	ET0_RX_DV hold time	$t_{RDVh}$	10	—	ns		
	ET0_ERXD0 to ET_ERXD3 setup time	$t_{MRDs}$	10	—	ns		
	ET0_ERXD0 to ET_ERXD3 hold time	$t_{MRDh}$	10	—	ns		
	ET0_RX_ER setup time	$t_{RERs}$	10	—	ns	Figure 52.70	
	ET0_RX_ER hold time	$t_{RESh}$	10	—	ns		
	ET0_WOL output delay	$t_{WOLd}$	1	23.5	ns		

Note: The following pins must use pins that have a letter appended to their name, for instance “\_A”, “\_B”, to indicate group membership. For the ETHERC (RMII) Host interface, the AC portion of the electrical characteristics is measured for each group. REF50CK0\_A, REF50CK0\_B, RMII0\_XXXX\_A, RMII0\_XXXX\_B.

Note 1. RMII\_TXD\_EN, RMII\_TXD1, RMII\_TXD0.

Note 2. RMII\_CRS\_DV, RMII\_RXD1, RMII\_RXD0, RMII\_RX\_ER.

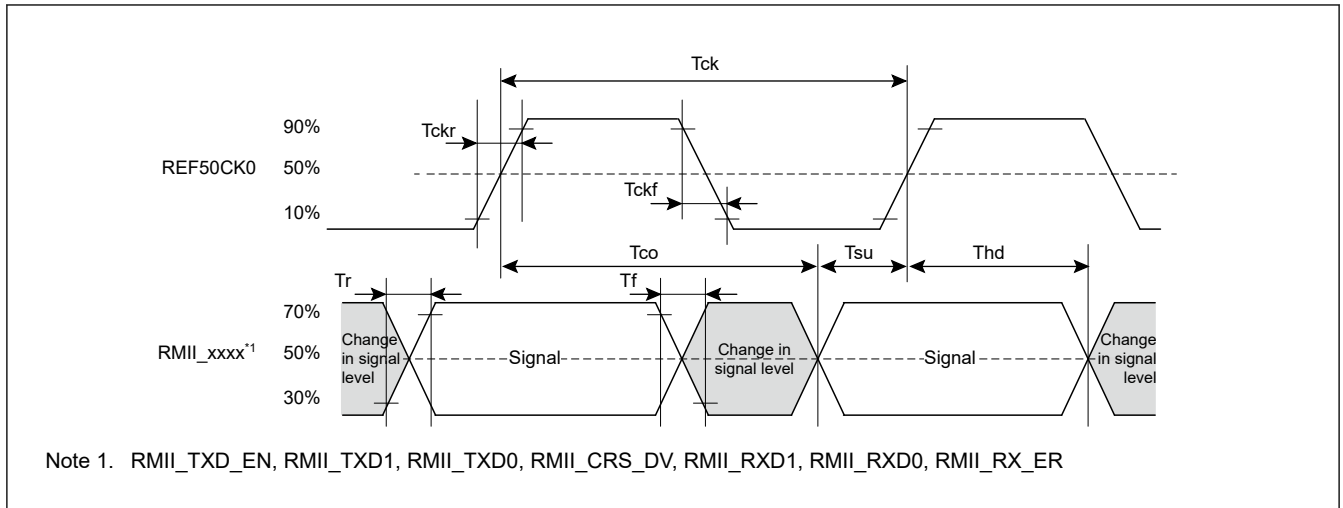


Figure 52.61 REF50CK0 and RMIIX signal timing

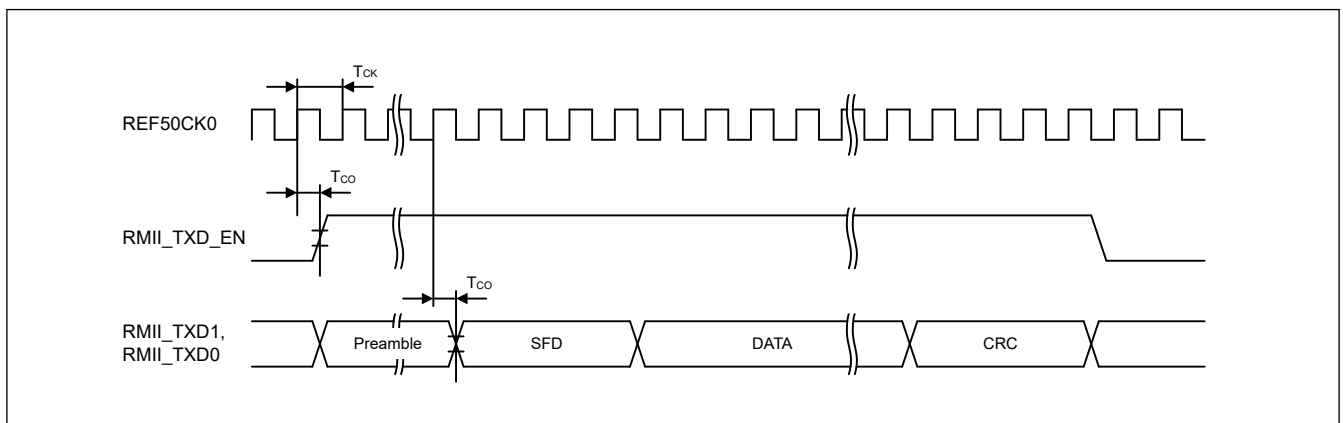


Figure 52.62 RMIIX transmission timing

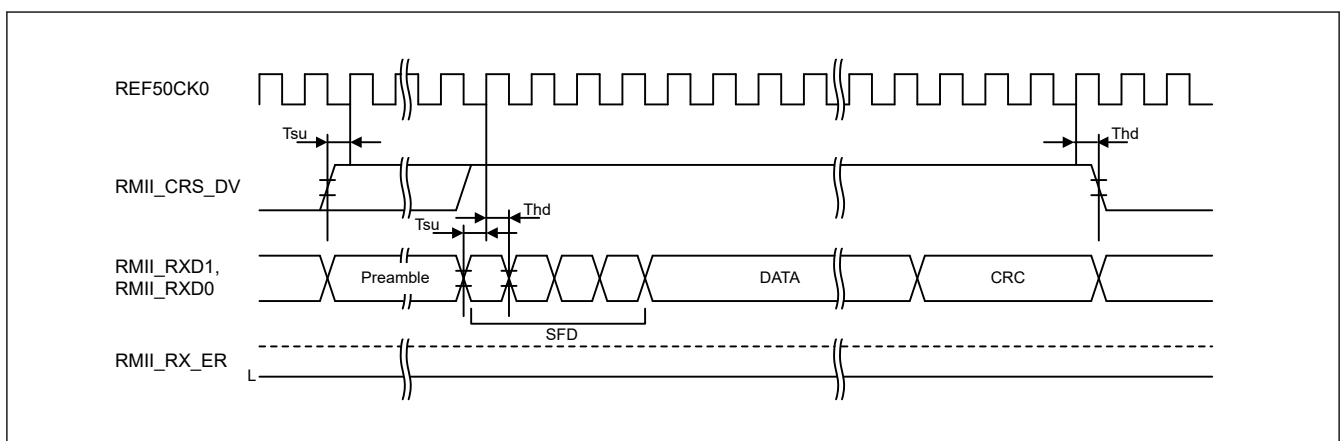


Figure 52.63 RMIIX reception timing in normal operation

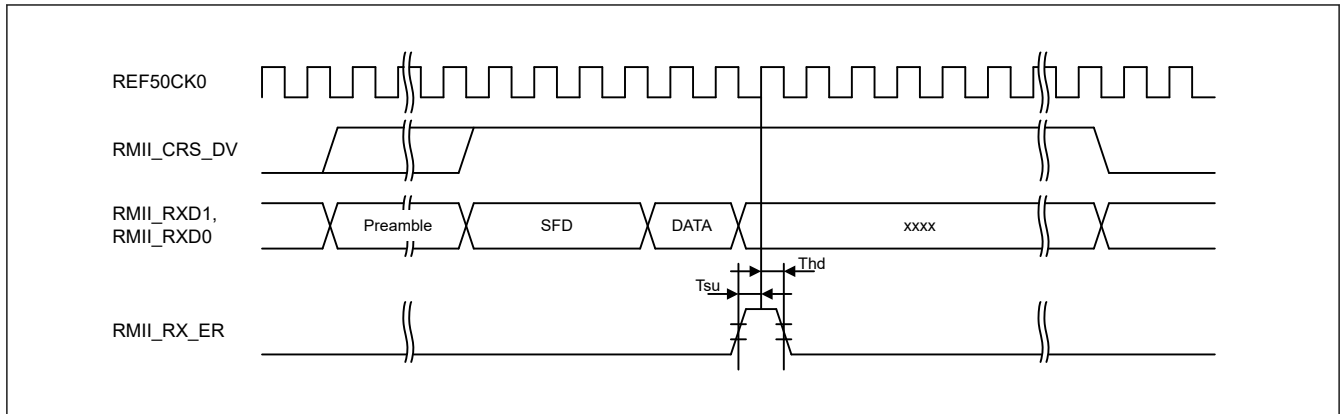


Figure 52.64 RMI reception timing when an error occurs

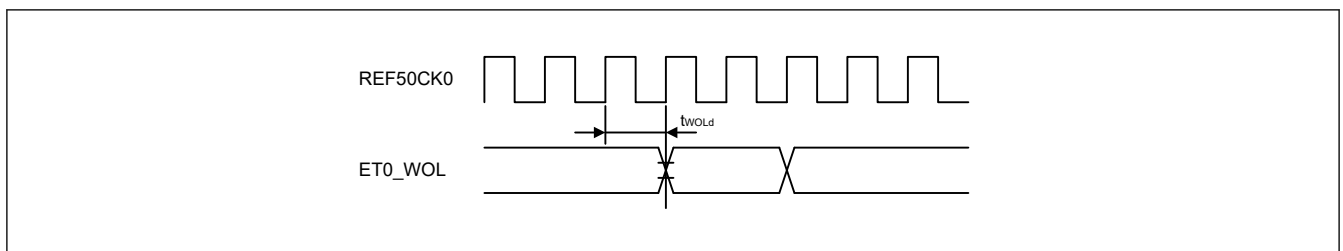


Figure 52.65 WOL output timing for RMI

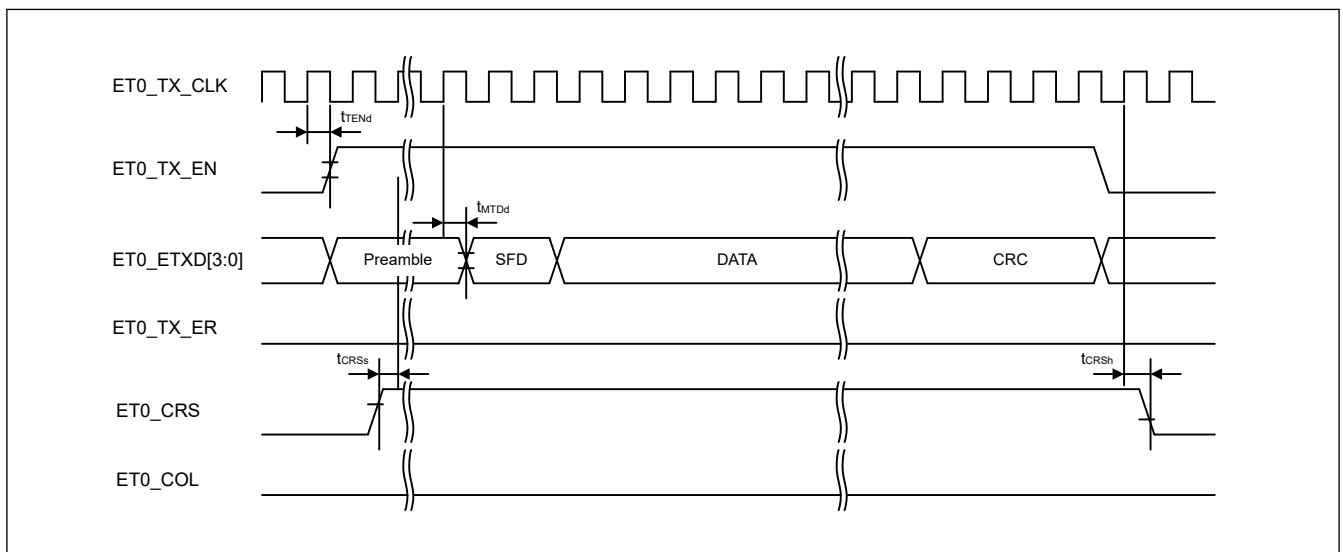


Figure 52.66 MII transmission timing in normal operation

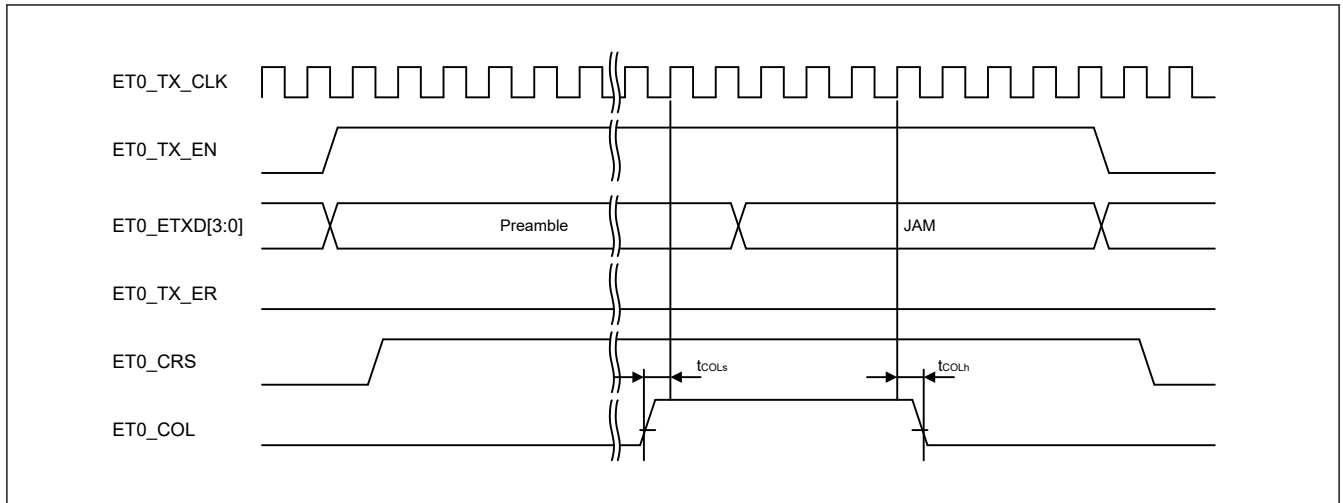


Figure 52.67 MII transmission timing when a conflict occurs

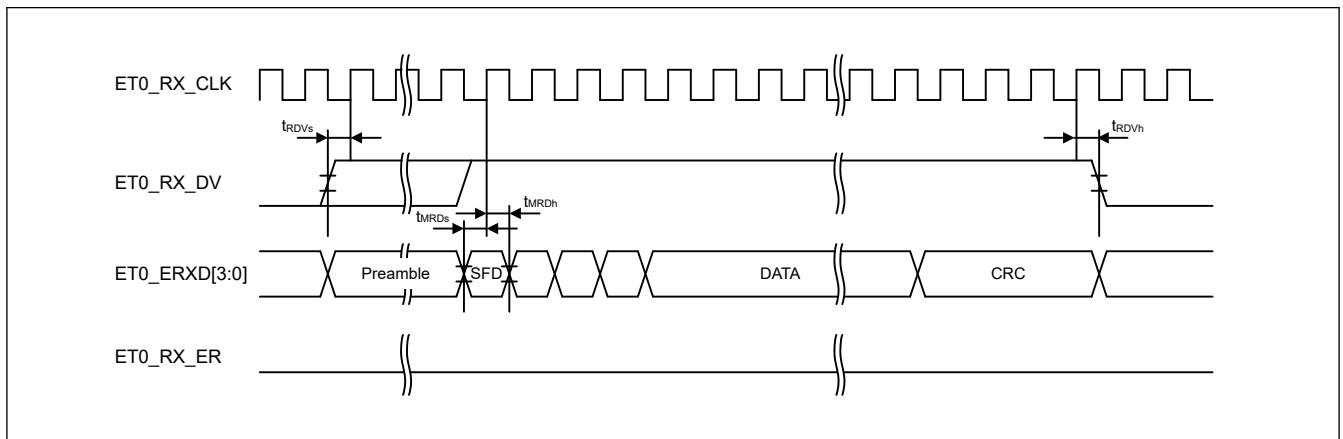


Figure 52.68 MII reception timing in normal operation

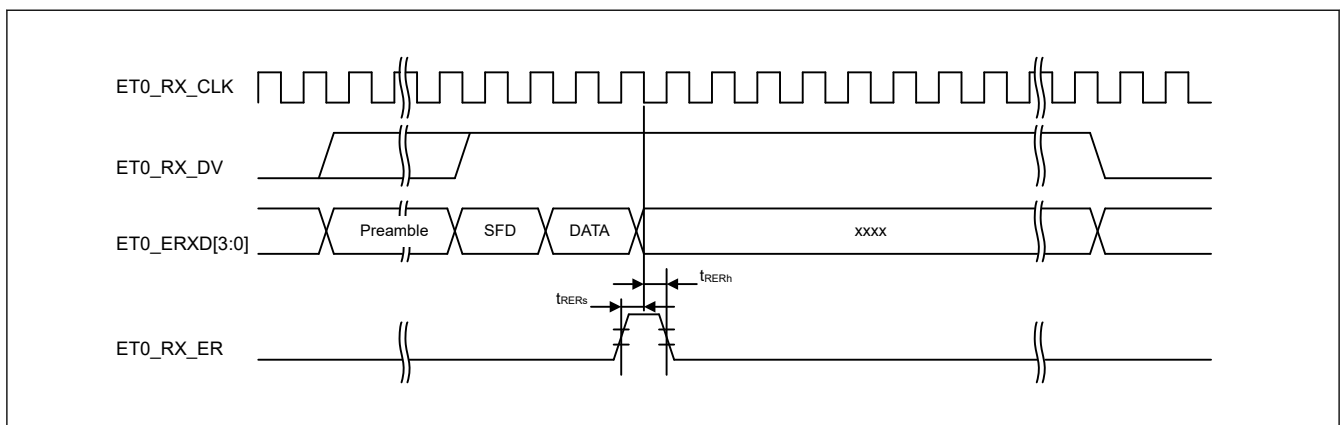


Figure 52.69 MII reception timing when an error occurs

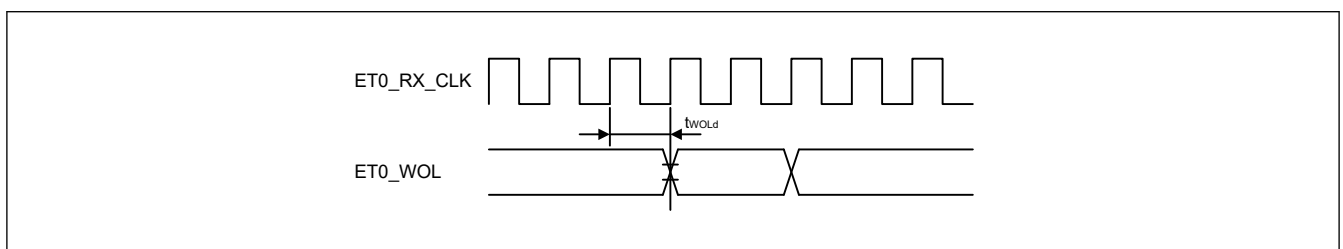


Figure 52.70 WOL output timing for MII

## 52.3.14 CEU Timing

Table 52.49 Capture Engine Unit Signal Timing

Parameter	Symbol	VCC	Min	Max	Unit	Test conditions
Vertical sync (VIO_VD) setup time (Camera clock rising)	$t_{VDS}$	2.70 V or above	2.0	—	ns	Figure 52.71 Figure 52.72
		1.68 V or above	4.5	—		
Vertical sync (VIO_VD) setup time (Camera clock falling)	$t_{VDS}$	2.70 V or above	2.5	—	ns	
		1.68 V or above	4.5	—		
Vertical sync (VIO_VD) hold time	$t_{VDH}$	2.70 V or above	3.5	—	ns	
		1.68 V or above	5.5	—		
Horizontal sync (VIO_HD) setup time (Camera clock rising)	$t_{VHDS}$	2.70 V or above	2.0	—	ns	
		1.68 V or above	4.5	—		
Horizontal sync (VIO_HD) setup time (Camera clock falling)	$t_{VHDS}$	2.70 V or above	2.5	—	ns	
		1.68 V or above	4.5	—		
Horizontal sync (VIO_HD) hold time	$t_{VHDH}$	2.70 V or above	3.5	—	ns	
		1.68 V or above	5.5	—		
Capture image data (VIO_D) setup time (Camera clock rising)	$t_{VDTS}$	2.70 V or above	2.0	—	ns	
		1.68 V or above	4.5	—		
Capture image data (VIO_D) setup time (Camera clock falling)	$t_{VDTS}$	2.70 V or above	2.5	—	ns	
		1.68 V or above	4.5	—		
Capture image data (VIO_D) hold time	$t_{VDTH}$	2.70 V or above	3.5	—	ns	
		1.68 V or above	5.5	—		
Camera clock cycle	$t_{VCYC}$	2.70 V or above	11.5	—	ns	
		1.68 V or above	23.0	—		
Camera clock high level width	$t_{VHW}$	2.70 V or above	$0.4 \times t_{VCYC}$	—	ns	
		1.68 V or above	$0.4 \times t_{VCYC}$	—		
Camera clock low level width	$t_{VLW}$	2.70 V or above	$0.4 \times t_{VCYC}$	—	ns	
		1.68 V or above	$0.4 \times t_{VCYC}$	—		
Field identification signal (VIO_FLD) setup time (Camera clock rising)	$t_{VFDS}$	2.70 V or above	2.0	—	ns	
		1.68 V or above	4.5	—		
Field identification signal (VIO_FLD) setup time (Camera clock falling)	$t_{VFDS}$	2.70 V or above	2.5	—	ns	
		1.68 V or above	4.5	—		
Field identification signal (VIO_FLD) hold time	$t_{VFDH}$	2.70 V or above	3.5	—	ns	
		1.68 V or above	5.5	—		



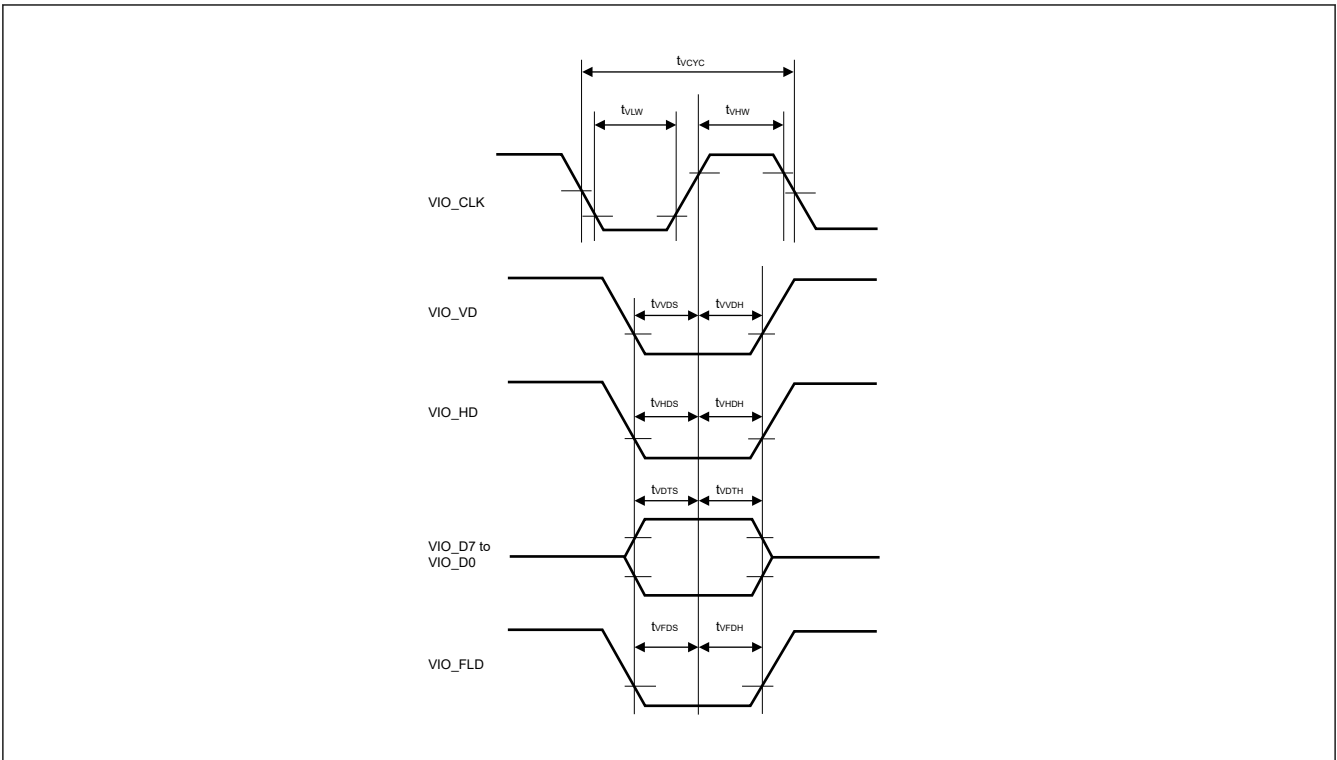


Figure 52.71 Capture Engine Unit Module Signal Timing of data capturing on the rising edge of VIO\_CLK

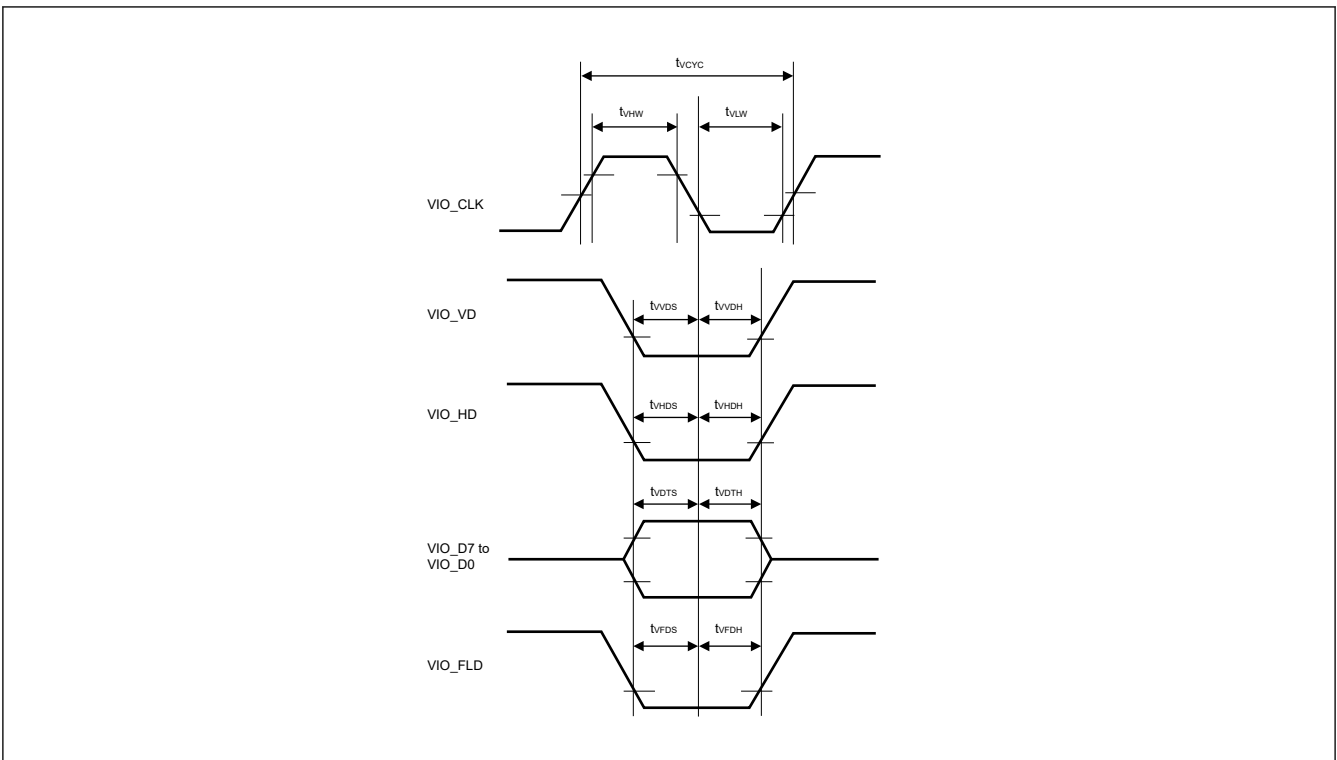


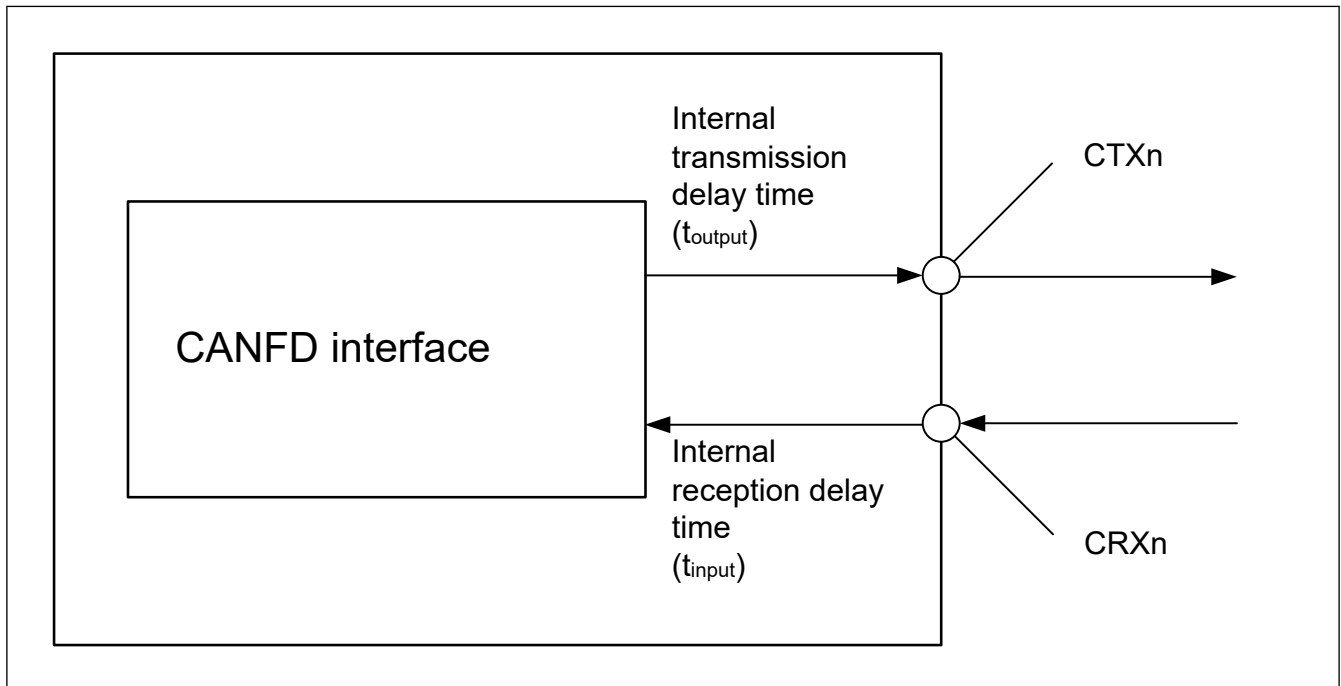
Figure 52.72 Capture Engine Unit Module Signal Timing of data capturing on the falling edge of VIO\_CLK

### 52.3.15 CANFD Timing

**Table 52.50 CANFD interface timing**

Parameter	Symbol	VCC/VCC2	Min	Max	Unit	Test conditions
Internal delay time	$t_{node}$	2.70 V or above	—	50	ns	Figure 52.73
		1.68 V or above (VCC) 1.65 V or above (VCC2)	—	50		

Note: Internal delay time ( $t_{node}$ ) = Internal transfer delay time ( $t_{output}$ ) + Internal receive delay time ( $t_{input}$ )



**Figure 52.73 CANFD interface condition**

## 52.4 USB Characteristics

### 52.4.1 USBFS Timing

**Table 52.51 USBFS low-speed characteristics for host only (USB\_DP and USB\_DM pin characteristics) (1 of 2)**

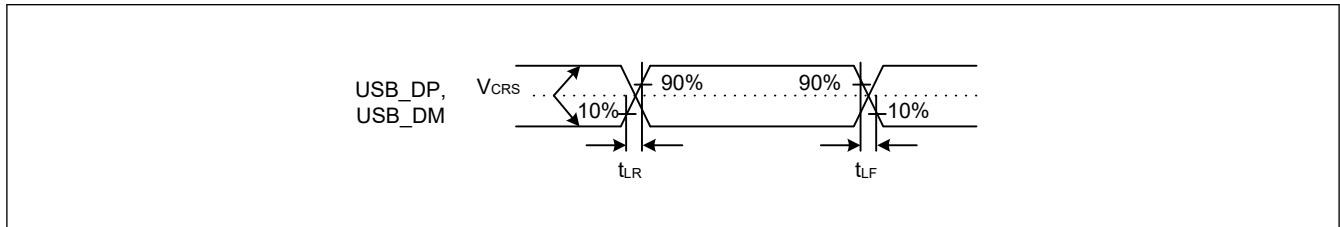
Conditions: VCC = VCC\_USB = 3.0 to 3.6 V, USBCLK = 48 MHz

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions	
Input characteristics	Input high voltage	$V_{IH}$	2.0	—	—	V	—
	Input low voltage	$V_{IL}$	—	—	0.8	V	—
	Differential input sensitivity	$V_{DI}$	0.2	—	—	V	USB_DP - USB_DM
	Differential common-mode range	$V_{CM}$	0.8	—	2.5	V	—
Output characteristics	Output high voltage	$V_{OH}$	2.8	—	3.6	V	$I_{OH} = -200 \mu A$
	Output low voltage	$V_{OL}$	0.0	—	0.3	V	$I_{OL} = 2 \text{ mA}$
	Cross-over voltage	$V_{CRS}$	1.3	—	2.0	V	Figure 52.74
	Rise time	$t_{LR}$	75	—	300	ns	
	Fall time	$t_{LF}$	75	—	300	ns	
	Rise/fall time ratio	$t_{LR} / t_{LF}$	80	—	125	%	$t_{LR} / t_{LF}$

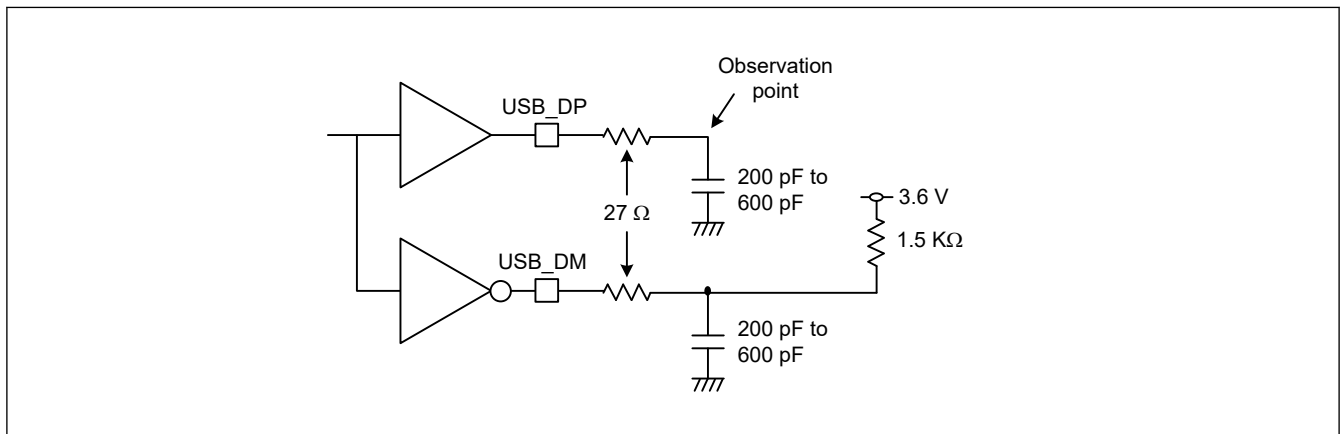
**Table 52.51 USBFS low-speed characteristics for host only (USB\_DP and USB\_DM pin characteristics) (2 of 2)**

Conditions: VCC = VCC\_USB = 3.0 to 3.6 V, USBCLK = 48 MHz

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions	
Pull-up and pull-down characteristics	USB_DP and USB_DM pull-down resistance in host controller mode	R <sub>pd</sub>	14.25	—	24.80	kΩ	—



**Figure 52.74 USB\_DP and USB\_DM output timing in low-speed mode**



**Figure 52.75 Test circuit in low-speed mode**

**Table 52.52 USBFS full-speed characteristics (USB\_DP and USB\_DM pin characteristics)**

Conditions: VCC = VCC\_USB = 3.0 to 3.6 V, USBCLK = 48 MHz

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions	
Input characteristics	Input high voltage	V <sub>IH</sub>	2.0	—	—	V	—
	Input low voltage	V <sub>IL</sub>	—	—	0.8	V	—
	Differential input sensitivity	V <sub>DI</sub>	0.2	—	—	V	USB_DP - USB_DM
	Differential common-mode range	V <sub>CM</sub>	0.8	—	2.5	V	—
Output characteristics	Output high voltage	V <sub>OH</sub>	2.8	—	3.6	V	I <sub>OH</sub> = -200 μA
	Output low voltage	V <sub>OL</sub>	0.0	—	0.3	V	I <sub>OL</sub> = 2 mA
	Cross-over voltage	V <sub>CRS</sub>	1.3	—	2.0	V	Figure 52.76
	Rise time	t <sub>LR</sub>	4	—	20	ns	t <sub>FR</sub> / t <sub>FF</sub>
	Fall time	t <sub>LF</sub>	4	—	20	ns	
	Rise/fall time ratio	t <sub>LR</sub> / t <sub>LF</sub>	90	—	111.11	%	
	Output resistance	Z <sub>DRV</sub>	28	—	44	Ω	USBFS: R <sub>s</sub> = 27 Ω included
Pull-up and pull-down characteristics	DM pull-up resistance in device controller mode	R <sub>pu</sub>	0.900	—	1.575	kΩ	During idle state
		R <sub>pu</sub>	1.425	—	3.090	kΩ	During transmission and reception
	USB_DP and USB_DM pull-down resistance in host controller mode	R <sub>pd</sub>	14.25	—	24.80	kΩ	—

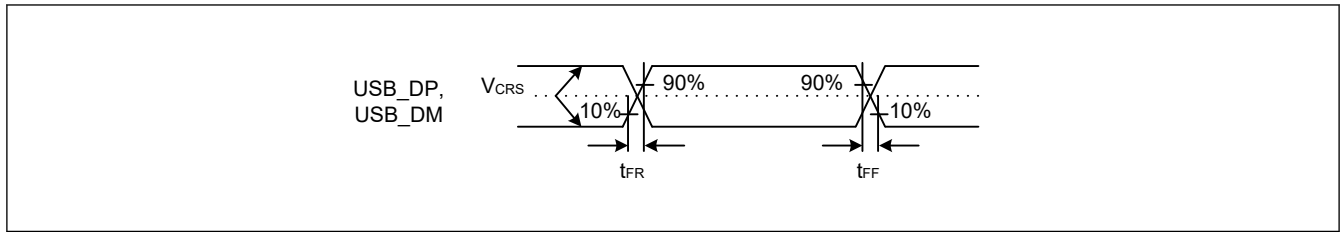


Figure 52.76 USB\_DP and USB\_DM output timing in full-speed mode

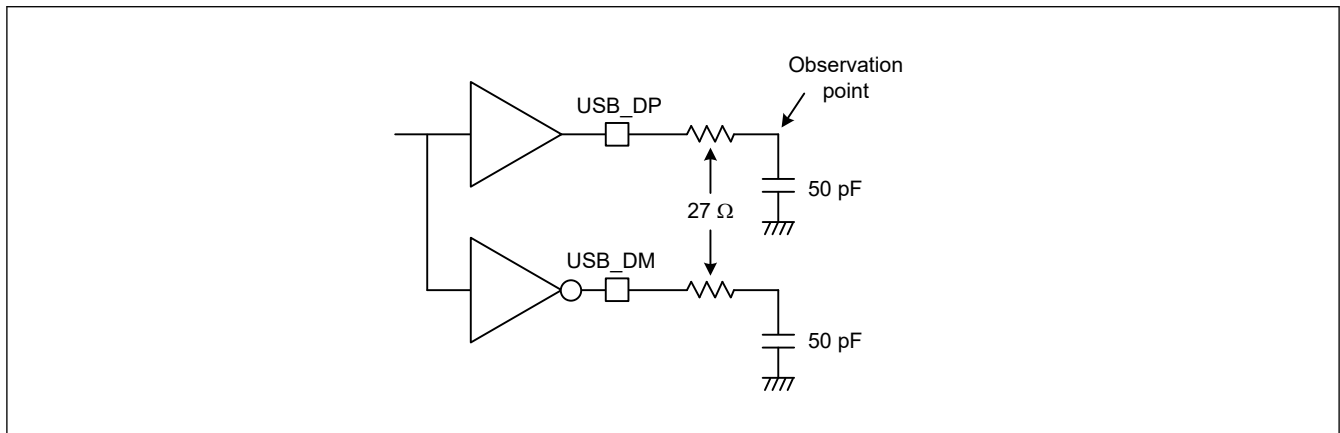


Figure 52.77 Test circuit in full-speed mode

### 52.5 ADC12 Characteristics

Table 52.53 A/D conversion characteristics for unit 0 (DCDC mode) (1 of 2)

Conditions: PCLKC = 1 to 60 MHz

Parameter			Min	Typ	Max	Unit	Test conditions
Frequency			1	—	60	MHz	—
Analog input capacitance			—	—	30	pF	—
Quantization error			—	±0.5	—	LSB	—
Resolution			—	—	12	Bits	—
Channel-dedicated sample-and-hold circuits in use (AN000, AN001)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	1.06 (0.4 + 0.25) <sup>2</sup>	—	—	μs	<ul style="list-style-type: none"> <li>Sampling of channel-dedicated sample-and-hold circuits in 24 states</li> <li>Sampling in 15 states</li> </ul>
	Offset error		—	±1.5	±3.5	LSB	AN000 to AN002 = VREFH0 - 0.25 V
	Full-scale error		—	±1.5	±3.5	LSB	AN000 to AN002 = VREFH0 - 0.25 V
	Absolute accuracy		—	±2.5	±10.5	LSB	LQFP package AVCC0 = 2.7 to 3.6V VREFH0 = 2.7V to AVCC0
			—	±2.5	±7.5		LQFP package AVCC0 = VREFH0 = 2.7 to 3.6V
	DNL differential nonlinearity error		—	±1.0	±2.0	LSB	—
	INL integral nonlinearity error		—	±1.5	±4.0	LSB	—
	Holding characteristics of sample-and hold circuits		—	—	20	μs	—
	Dynamic range		0.25	—	VREFH0 - 0.25	V	—

**Table 52.53 A/D conversion characteristics for unit 0 (DCDC mode) (2 of 2)**

Conditions: PCLKC = 1 to 60 MHz

Parameter			Min	Typ	Max	Unit	Test conditions
High-precision channels, Channel-dedicated sample-and-hold circuits not in use (AN000 to AN002, AN004 to AN008)	Conversion time* <sup>1</sup> (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.48 (0.267) <sup>*2</sup>	—	—	μs	Sampling in 16 states
		Max. = 400 Ω	0.40 (0.183) <sup>*2</sup>	—	—	μs	Sampling in 11 states AVCC0 = VREFH0 = 3.0 to 3.6 V
	Offset error		—	±1.0	±2.5	LSB	—
	Full-scale error		—	±1.0	±3.5	LSB	—
	Absolute accuracy		—	±2.0	±7.5	LSB	LQFP package AVCC0 = 2.7 to 3.6V VREFH0 = 2.7V to AVCC0
			—	±2.0	±6.0		LQFP package AVCC0 = VREFH0 = 2.7 to 3.6V
	DNL differential nonlinearity error		—	±0.5	±2.0	LSB	—
INL integral nonlinearity error		—	±1.0	±2.5	LSB	—	

Note: These specification values apply when only one A/D is operating and D/A and ACMPHS are not operating during A/D conversion. If other A/D, D/A, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.

The use of ports 0 as digital outputs is not allowed when the 12-Bit A/D converter is used.

The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and 12-bit A/D converter input voltage is stable.

Note 1. The conversion time includes the sampling and comparison times. The number of sampling states is indicated for the test conditions.

Note 2. Values in parentheses indicate the sampling time.

**Table 52.54 A/D conversion characteristics for unit 1 (DCDC mode)**

Conditions: PCLKC = 1 to 60 MHz

Parameter			Min	Typ	Max	Unit	Test conditions
Frequency			1	—	60	MHz	—
Analog input capacitance			—	—	30	pF	—
Quantization error			—	±0.5	—	LSB	—
Resolution			—	—	12	Bits	—
High-precision channels (AN100 to AN102, AN104, AN105)	Conversion time* <sup>1</sup> (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.48 (0.267) <sup>*2</sup>	—	—	μs	Sampling in 16 states
		Max. = 400 Ω	0.40 (0.183) <sup>*2</sup>	—	—	μs	Sampling in 11 states AVCC0 = VREFH = 3.0 to 3.6 V
	Offset error		—	±1.0	±2.5	LSB	—
	Full-scale error		—	±1.0	±3.5	LSB	—
	Absolute accuracy		—	±2.0	±7.5	LSB	LQFP package AVCC0 = 2.7 to 3.6V VREFH = 2.7V to AVCC0
			—	±2.0	±6.0		LQFP package AVCC0 = VREFH = 2.7 to 3.6V
	DNL differential nonlinearity error		—	±0.5	±2.0	LSB	—
INL integral nonlinearity error		—	±1.0	±2.5	LSB	—	

Note: These specification values apply when only one A/D is operating and D/A and ACMPHS are not operating during A/D conversion. If other A/D, D/A, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.

The use of ports 0 as digital outputs is not allowed when the 12-Bit A/D converter is used.

The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and 12-bit A/D converter input voltage are stable.

Note 1. The conversion time includes the sampling and comparison times. The number of sampling states is indicated for the test conditions.

Note 2. Values in parentheses indicate the sampling time.

**Table 52.55 A/D conversion characteristics for unit 0 (External VDD mode)**

Conditions: PCLKC = 1 to 60 MHz

AVCC0 = 2.7 to 3.6 V, VREFH0 = 2.7 to 3.6 V

Parameter			Min	Typ	Max	Unit	Test conditions
Frequency			1	—	60	MHz	—
Analog input capacitance			—	—	30	pF	—
Quantization error			—	±0.5	—	LSB	—
Resolution			—	—	12	Bits	—
Channel-dedicated sample-and-hold circuits in use (AN000, AN001)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	1.06 (0.4 + 0.25)*2	—	—	μs	<ul style="list-style-type: none"> <li>Sampling of channel-dedicated sample-and-hold circuits in 24 states</li> <li>Sampling in 15 states</li> </ul>
	Offset error		—	±1.5	±3.5	LSB	AN000 to AN002 = 0.25 V
	Full-scale error		—	±1.5	±3.5	LSB	AN000 to AN002 = VREFH0 - 0.25 V
	Absolute accuracy		—	±2.5	±5.5	LSB	—
	DNL differential nonlinearity error		—	±1.0	±2.0	LSB	—
	INL integral nonlinearity error		—	±1.5	±3.0	LSB	—
	Holding characteristics of sample-and hold circuits		—	—	20	μs	—
Dynamic range			0.25	—	VREFH0 - 0.25	V	—
High-precision channels, Channel-dedicated sample-and-hold circuits not in use (AN000 to AN002, AN004 to AN008)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.48 (0.267)*2	—	—	μs	Sampling in 16 states
		Max. = 400 Ω	0.40 (0.183)*2	—	—	μs	Sampling in 11 states AVCC0 = 3.0 to 3.6 V 3.0 V ≤ VREFH0 ≤ AVCC0
	Offset error		—	±1.0	±2.5	LSB	—
	Full-scale error		—	±1.0	±3.5	LSB	—
	Absolute accuracy		—	±2.0	±4.5	LSB	—
	INL integral nonlinearity error		—	±1.0	±2.5	LSB	—

Note: These specification values apply when only one A/D is operating and D/A and ACMPHS are not operating during A/D conversion. If other A/D, D/A, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.

The use of ports 0 as digital outputs is not allowed when the 12-Bit A/D converter is used.

The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and 12-bit A/D converter input voltage is stable.

Note 1. The conversion time includes the sampling and comparison times. The number of sampling states is indicated for the test conditions.

Note 2. Values in parentheses indicate the sampling time.

**Table 52.56 A/D conversion characteristics for unit 1 (External VDD mode) (1 of 2)**

Conditions: PCLKC = 1 to 60 MHz

AVCC0 = 2.7 to 3.6 V, VREFH = 2.7 to 3.6 V

Parameter	Min	Typ	Max	Unit	Test conditions
Frequency	1	—	60	MHz	—
Analog input capacitance	—	—	30	pF	—

**Table 52.56 A/D conversion characteristics for unit 1 (External VDD mode) (2 of 2)**

Conditions: PCLKC = 1 to 60 MHz  
 AVCC0 = 2.7 to 3.6 V, VREFH = 2.7 to 3.6 V

Parameter		Min	Typ	Max	Unit	Test conditions	
Quantization error		—	±0.5	—	LSB	—	
Resolution		—	—	12	Bits	—	
High-precision channels (AN100 to AN102, AN104, AN105)	Conversion time*1 (operation at PCLKC = 60 MHz)	Permissible signal source impedance Max. = 1 kΩ	0.48 (0.267)*2	—	—	μs	Sampling in 16 states
		Max. = 400 Ω	0.40 (0.183)*2	—	—	μs	Sampling in 11 states AVCC0 = 3.0 to 3.6 V 3.0 V ≤ VREFH ≤ AVCC0
	Offset error		—	±1.0	±2.5	LSB	—
	Full-scale error		—	±1.0	±3.5	LSB	—
	Absolute accuracy		—	±2.0	±4.5	LSB	—
	DNL differential nonlinearity error		—	±0.5	±1.5	LSB	—
	INL integral nonlinearity error		—	±1.0	±2.5	LSB	—

Note: These specification values apply when only one A/D is operating and D/A and ACMPHS are not operating during A/D conversion. If other A/D, D/A, or ACMPHS is operating or bus access occurs during A/D conversion, values might not fall within the indicated ranges.

The use of ports 0 as digital outputs is not allowed when the 12-Bit A/D converter is used.

The characteristics apply when AVCC0, AVSS0, VREFH0, VREFH, VREFL0, VREFL, and 12-bit A/D converter input voltage are stable.

Note 1. The conversion time includes the sampling and comparison times. The number of sampling states is indicated for the test conditions.

Note 2. Values in parentheses indicate the sampling time.

**Table 52.57 A/D internal reference voltage characteristics**

Parameter	Min	Typ	Max	Unit	Test conditions
A/D internal reference voltage	1.13	1.18	1.28	V	—
Sampling time	4.15	—	—	μs	—

For the characteristics of VBATT 1/3 voltage monitor, see [section 52.10. VBATT Characteristics](#).

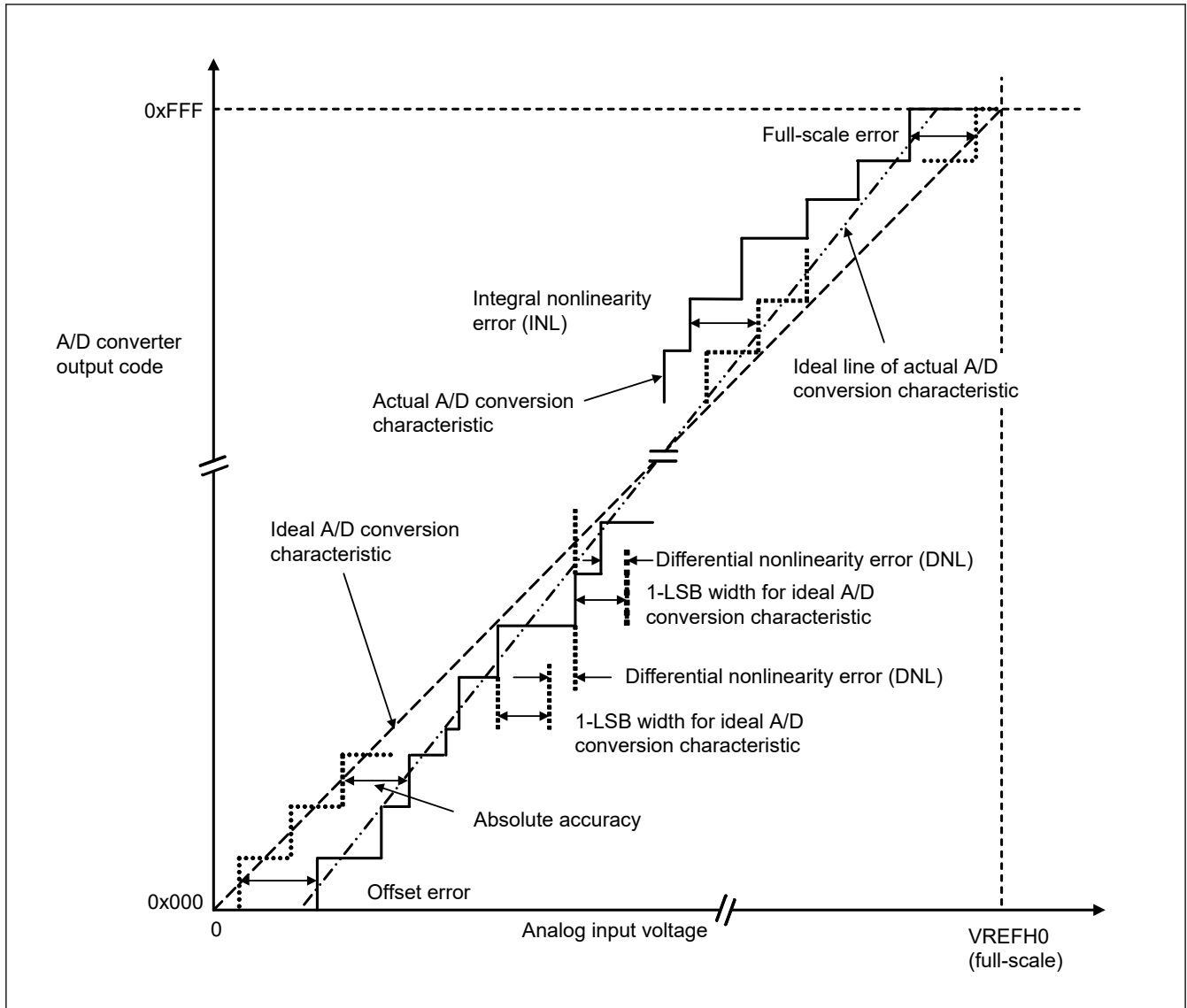


Figure 52.78 Illustration of ADC12 characteristic terms

### Absolute accuracy

Absolute accuracy is the difference between output code based on the theoretical A/D conversion characteristics, and the actual A/D conversion result. When measuring absolute accuracy, the voltage at the midpoint of the width of the analog input voltage (1-LSB width), which can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as an analog input voltage. For example, if 12-bit resolution is used and the reference voltage  $V_{REFH0} = 3.072$  V, then the 1-LSB width becomes 0.75 mV, and 0 mV, 0.75 mV, and 1.5 mV are used as the analog input voltages. If the analog input voltage is 6 mV, an absolute accuracy of  $\pm 5$  LSB means that the actual A/D conversion result is in the range of 0x003 to 0x00D, though an output code of 0x008 can be expected from the theoretical A/D conversion characteristics.

### Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

### Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between the 1-LSB width based on the ideal A/D conversion characteristics and the width of the actual output code.

### Offset error

Offset error is the difference between the transition point of the ideal first output code and the actual first output code.



**Full-scale error**

Full-scale error is the difference between the transition point of the ideal last output code and the actual last output code.

**52.6 DAC12 Characteristics****Table 52.58 D/A conversion characteristics**

Parameter	Min	Typ	Max	Unit	Test conditions	
Resolution	—	—	12	Bits	—	
Without output amplifier (for pin output, AVCC0 ≥ 1.65V)						
Absolute accuracy	VREFH ≥ 2.7V	—	—	±24	LSB	Resistive load 2 MΩ
	VREFH < 2.7V	—	—	±36		
INL	VREFH ≥ 2.7V	—	±2.0	±8.0	LSB	Resistive load 2 MΩ
	VREFH < 2.7V	—	±2.0	±8.0		
DNL	VREFH ≥ 2.7V	—	±1.0	±2.0	LSB	—
	VREFH < 2.7V	—	±1.0	±3.0		
Output impedance	—	8.5	—	kΩ	—	
Conversion time	VREFH ≥ 2.7V	—	—	3.0	μs	Resistive load 2 MΩ, Capacitive load 20 pF
	VREFH < 2.7V	—	—	6.0		
Output voltage range	0	—	VREFH	V	—	
Without output amplifier (for internal output, AVCC0 ≥ 1.65V)						
Absolute accuracy	VREFH ≥ 2.7V	—	—	±4.0	LSB	—
	VREFH < 2.7V	—	—	±6.0		
Conversion time	VREFH ≥ 2.7V	—	—	3.0	μs	—
	VREFH < 2.7V	—	—	6.0		
Output voltage range	0	—	VREFH	V	—	
With output amplifier (AVCC0 ≥ 2.70V)						
INL	—	±2.0	±4.0	LSB	—	
DNL	—	±1.0	±2.0	LSB	—	
Conversion time	—	—	3.5	μs	—	
Resistive load	5	—	—	kΩ	—	
Capacitive load	—	—	50	pF	—	
Output voltage range	VREFH ≥ 2.7V	0.20	—	VREFH – 0.20	V	—
	VREFH < 2.7V	0.22	—	VREFH – 0.22		—

### 52.7 TSN Characteristics

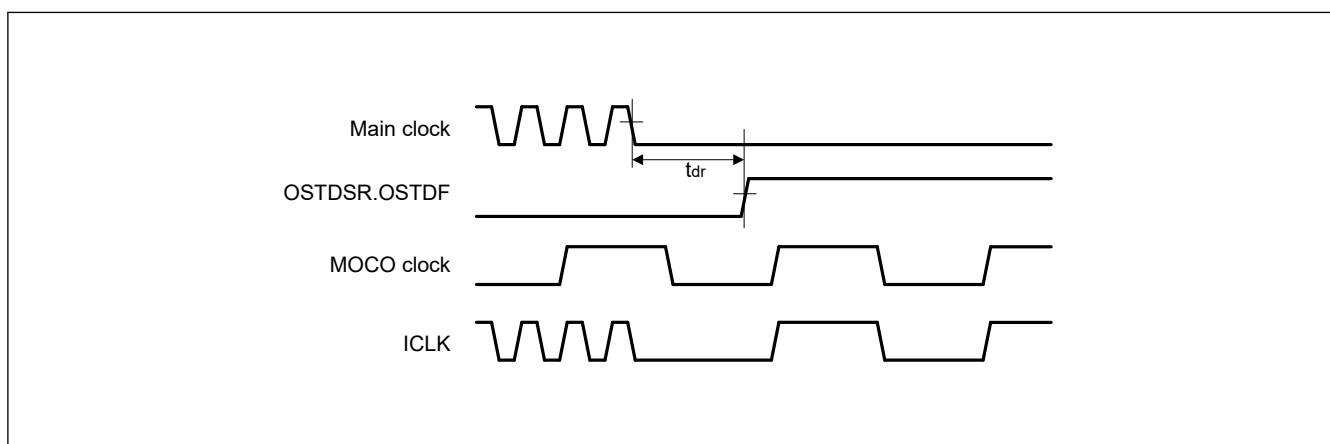
**Table 52.59 TSN characteristics**

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Relative accuracy	—	—	± 1.0	—	°C	—
Temperature slope	—	—	4.0	—	mV/°C	—
Output voltage (at 25 °C)	—	—	1.24	—	V	—
Temperature sensor start time	t <sub>START</sub>	—	—	30	μs	—
Sampling time	—	4.15	—	—	μs	—

### 52.8 OSC Stop Detect Characteristics

**Table 52.60 Oscillation stop detection circuit characteristics**

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Detection time	t <sub>dr</sub>	—	—	1	ms	Figure 52.79



**Figure 52.79 Oscillation stop detection timing**

## 52.9 POR and PVD Characteristics

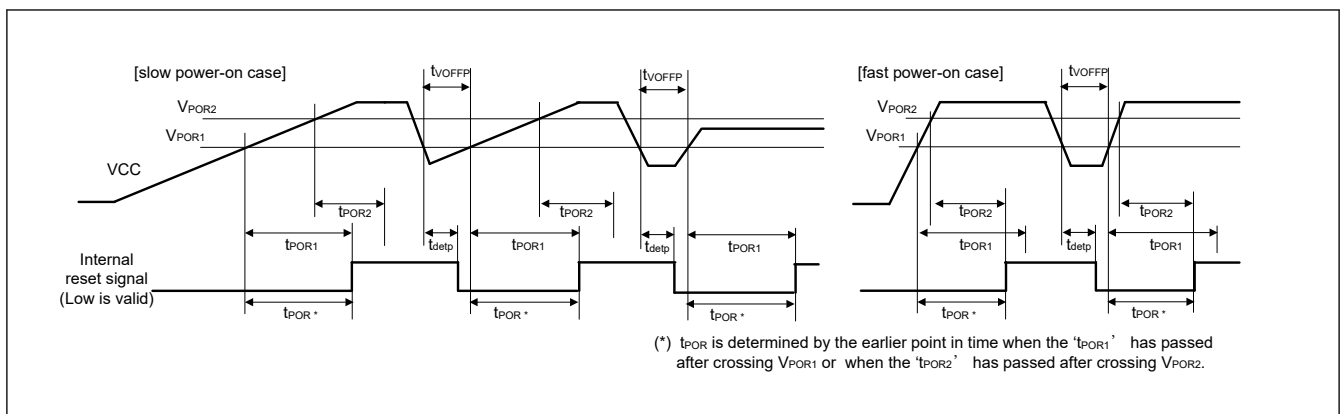
Table 52.61 Power-on reset circuit and voltage detection circuit characteristics (1 of 2)

Parameter		Symbol	Min	Typ	Max	Unit	Test conditions	
Voltage detection level	Power-on reset (POR)	Tj = 25°C	V <sub>POR1</sub>	1.55	1.60	1.68	V	Figure 52.80
				Tj = 105°C	1.55	1.60		
		Tj = 25°C	V <sub>POR2</sub>	1.65	1.70	1.79		
				Tj = 105°C	1.65	1.70		
	Voltage detection circuit (PVD0)	V <sub>det0_0</sub>	2.76	2.85	2.99	Figure 52.81		
		V <sub>det0_1</sub>	2.50	2.58	2.71			
		V <sub>det0_2</sub>	2.08	2.15	2.27			
		V <sub>det0_3</sub>	1.94	2.00	2.12			
		V <sub>det0_4</sub>	1.84	1.90	2.01			
		V <sub>det0_5</sub>	1.74	1.80	1.91			
		V <sub>det0_6</sub>	1.65	1.70	1.81			
		V <sub>det0_7</sub>	1.55	1.60	1.70			
	Voltage detection circuit (PVDn) (n = 1, 2)	V <sub>detn_3_rise</sub>	3.78	3.92	4.10	Figure 52.82		
		V <sub>detn_3_fall</sub>	3.72	3.86	4.04			
		V <sub>detn_4_rise</sub>	3.09	3.20	3.35			
		V <sub>detn_4_fall</sub>	3.03	3.14	3.29			
		V <sub>detn_5_rise</sub>	3.05	3.16	3.31			
		V <sub>detn_5_fall</sub>	2.99	3.10	3.25			
		V <sub>detn_6_rise</sub>	3.03	3.14	3.29			
		V <sub>detn_6_fall</sub>	2.97	3.08	3.23			
V <sub>detn_7_rise</sub>		2.81	2.91	3.05				
V <sub>detn_7_fall</sub>		2.75	2.85	2.99				
V <sub>detn_8_rise</sub>		2.79	2.89	3.03				
V <sub>detn_8_fall</sub>		2.73	2.83	2.97				
V <sub>detn_9_rise</sub>		2.76	2.86	3.00				
V <sub>detn_9_fall</sub>		2.70	2.80	2.94				
V <sub>detn_10_rise</sub>		2.58	2.67	2.80				
V <sub>detn_10_fall</sub>		2.53	2.62	2.75				
Voltage detection level	Voltage detection circuit (PVDn) (n = 1, 2)	V <sub>detn_11_rise</sub>	2.30	2.38	2.51	V	Figure 52.82	
		V <sub>detn_11_fall</sub>	2.25	2.33	2.46			
		V <sub>detn_12_rise</sub>	1.88	1.94	2.05			
		V <sub>detn_12_fall</sub>	1.84	1.90	2.01			
		V <sub>detn_13_rise</sub>	1.84	1.90	2.01			
		V <sub>detn_13_fall</sub>	1.80	1.86	1.97			
		V <sub>detn_14_rise</sub>	1.72	1.78	1.89			
		V <sub>detn_14_fall</sub>	1.68	1.74	1.85			
Voltage detection level	Voltage detection circuit (PVDn) (n = 1, 2)	V <sub>detn_15_rise</sub>	1.69	1.75	1.85	V	Figure 52.82	
		V <sub>detn_15_fall</sub>	1.65	1.71	1.81			

**Table 52.61 Power-on reset circuit and voltage detection circuit characteristics (2 of 2)**

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions		
Internal reset time* <sup>1</sup>	Power-on reset time	$t_{POR1}$	—	—	8.2	ms	Figure 52.80	
		$t_{POR2}$	—	—	4.5		Figure 52.80	
	PVD0 reset time	$t_{PVD0}$	—	—	*1		Figure 52.80	
	PVD1 reset time	$t_{PVD1}$	—	—	*1		Figure 52.81	
	PVD2 reset time	$t_{PVD2}$	—	—	*1			
Minimum VCC down time (POR)* <sup>2</sup>	100mV < VD	$t_{VOFFP}$	500	—	—	μs	Figure 52.80	
	50mV < VD ≤ 100mV		900	—	—			
	VD ≤ 50mV		2000	—	—			
Minimum VCC down time (PVD)* <sup>2</sup>	PVD0 (OFS1(_SEC).PVDLPSEL = 0 in Deep Software Standby mode 1, 2)	$t_{VOFF}$	400	—	—	μs	Figure 52.81	
	PVD0 (Other than above), PVD1, PVD2		200	—	—			
Response delay time (POR)	100mV < VD	$t_{detp}$	—	—	500	μs	Figure 52.80	
	50mV < VD ≤ 100mV		—	—	900			
	VD ≤ 50mV		—	—	2000			
Response delay time (PVD)	PVD0 (OFS1(_SEC).PVDLPSEL = 0 in Deep Software Standby mode 1, 2)	50mV < VD	$t_{det}$	—	—	μs	Figure 52.81, Figure 52.82	
		50mV ≥ VD		—	—			200
	PVD0 (Other than above), PVD1, PVD2	100mV < VD		—	—			10
		100mV ≥ VD		—	—			200
PVD operation stabilization time (after PVD is enabled)	$T_d (E-A)$	—	—	20	μs	Figure 52.82		

- Note 1. The maximum value of  $t_{PVD0}$  is equal to  $t_{DSBY}$  because the internal reset time is maximized when returning from Deep Software Standby mode.  
 The maximum value of  $t_{PVD1}$ ,  $t_{PVD2}$  are equal to  $t_{STBY}$  because the internal reset time is maximized when returning from Deep Software Standby mode.
- Note 2. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels  $V_{POR1}$ ,  $V_{det1}$ , and  $V_{det2}$  for the POR / PVD.



**Figure 52.80 Power-on reset timing**

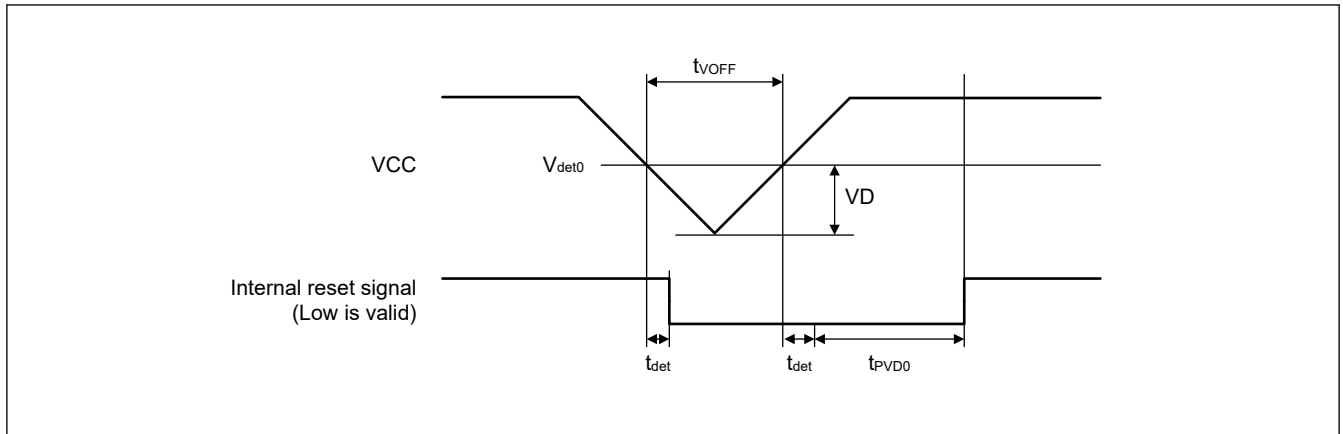


Figure 52.81 Voltage detection circuit timing ( $V_{det0}$ )

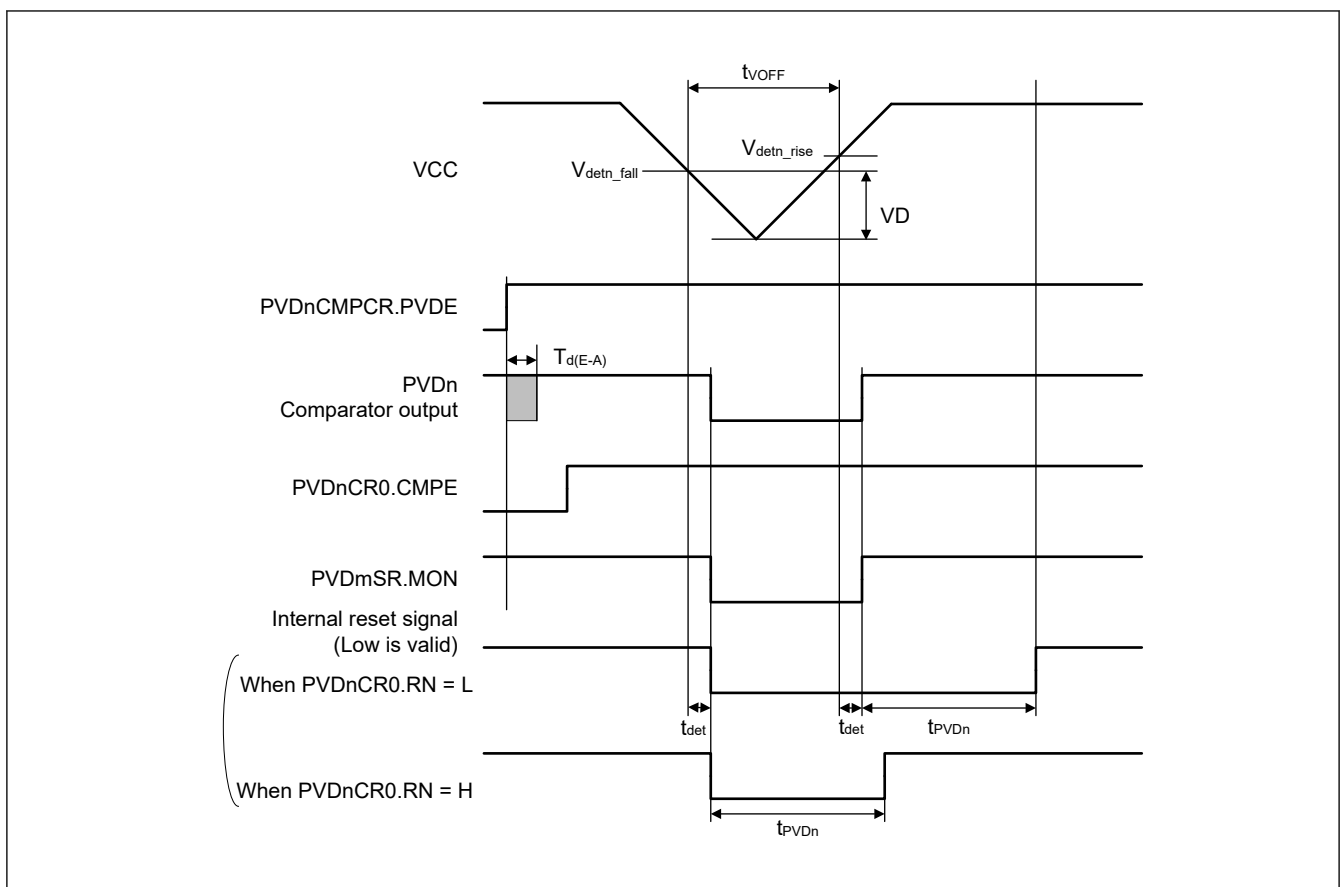


Figure 52.82 Voltage detection circuit timing ( $V_{detn}$ ) ( $n = 1, 2$ )

## 52.10 VBATT Characteristics

**Table 52.62 Battery backup function characteristics**

Conditions: VCC = VCC\_DCDC = VCC\_USB = 1.68 to 3.6 V, VBATT = 1.62 to 3.6 V

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Voltage level for switching to battery backup OFS1(_SEC).PVDAS and PVDLPSEL are 0 in Deep Software Standby mode 1, 2 (VDETVATT_n follows VDSEL[2:0] setting for PVD0)	V <sub>DETBATT_0</sub>	2.760	2.850	2.990	V	Figure 52.83
	V <sub>DETBATT_1</sub>	2.500	2.580	2.710		
	V <sub>DETBATT_2</sub>	2.080	2.150	2.270		
	V <sub>DETBATT_3</sub>	1.940	2.000	2.120		
	V <sub>DETBATT_4</sub>	1.840	1.900	2.010		
	V <sub>DETBATT_5</sub>	1.740	1.800	1.910		
	V <sub>DETBATT_6</sub>	1.650	1.700	1.810		
Voltage level for switching to battery backup (Other than above)	V <sub>DETBATT_0</sub>	2.710	2.800	2.940	V	
	V <sub>DETBATT_1</sub>	2.450	2.530	2.660		
	V <sub>DETBATT_2</sub>	2.030	2.100	2.220		
	V <sub>DETBATT_3</sub>	1.855	1.950	2.065		
	V <sub>DETBATT_4</sub>	1.790	1.850	1.960		
	V <sub>DETBATT_5</sub>	1.690	1.750	1.860		
VCC drop detection stabilization wait time*2	t <sub>DETWT</sub>	—	—	20	μs	
Lower-limit VBATT voltage for power supply switching caused by VCC voltage drop	V <sub>BATTSW</sub>	2.0	—	—	V	Figure 52.83
VCC-off period for starting power supply switching*1 (OFS1(_SEC).PVDAS and PVDLPSEL are 0 in Deep Software Standby mode 1, 2)	t <sub>VOFFBATT</sub>	400	—	—	μs	
VCC-off period for starting power supply switching*1 (Other than above)		200	—	—		
Backup domain power-down detection level	V <sub>PDR</sub> (BATR)	1.45	1.50	1.60	V	Figure 52.84
Time delay in assertion of the reset signal for the backup domain*3	t <sub>p</sub> (PDR <sub>L</sub> )	—	—	2000	μs	
Time delay in negation of the reset signal for the backup domain	t <sub>p</sub> (PDR <sub>H</sub> )	—	—	3000		
VBATT monitor operation stabilization time (after VBATTMNSLR.VBTMNSLR is changed to 1)	t <sub>MONWT</sub>	—	—	4.2	μs	—
VBATT voltage monitor level	V <sub>MONBATT</sub>	—	VBATT / 3	—	V	—
VBATT current increase (when VBATTMNSLR.VBTMNSLR is 1 compared to the case that VBATTMNSLR.VBTMNSLR is 0)	I <sub>VBATTSELB</sub>	—	1.50	2.35	μA	—
VCC current increase (when VBATTMNSLR.VBTMNSLR is 1 compared to the case that VBATTMNSLR.VBTMNSLR is 0)	I <sub>VBATTSELC</sub>	—	330	577	μA	—

Note 1. The VCC-off period for starting power supply switching indicates the period in which VCC is below the minimum value of the voltage level for switching to battery backup (V<sub>DETBATT</sub>).

In addition, this period indicates the time t<sub>VOFFP</sub> when VCC is below the minimum value of voltage detection levels V<sub>POR1</sub>.

Note 2. Stable time when VBTBPCR2.VDETLVL is changed or VBTBPCR2.VDETLVL is changed from 0 to 1.

Note 3. When the VBATT\_R recovers within this period, the backup domain reset signal may not be generated.

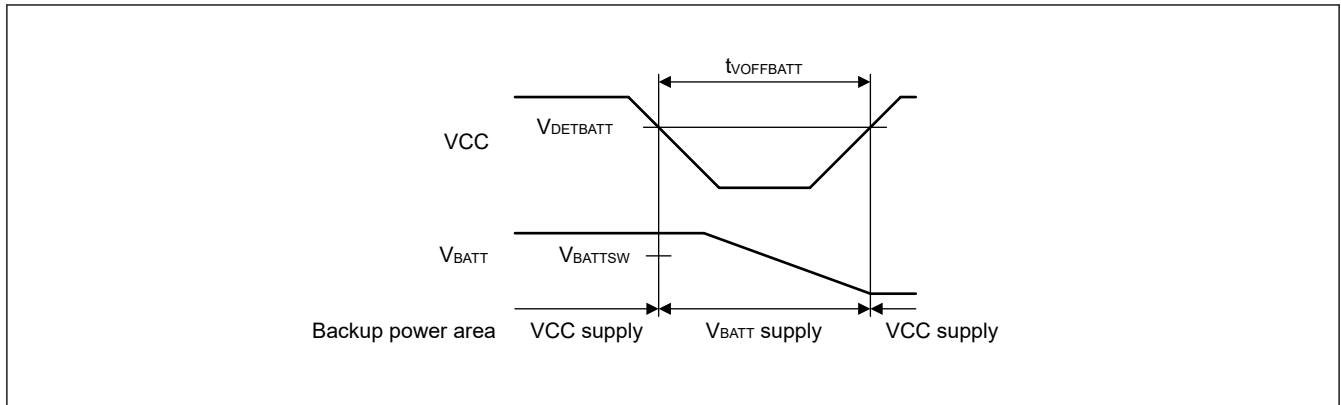


Figure 52.83 Battery backup function characteristics

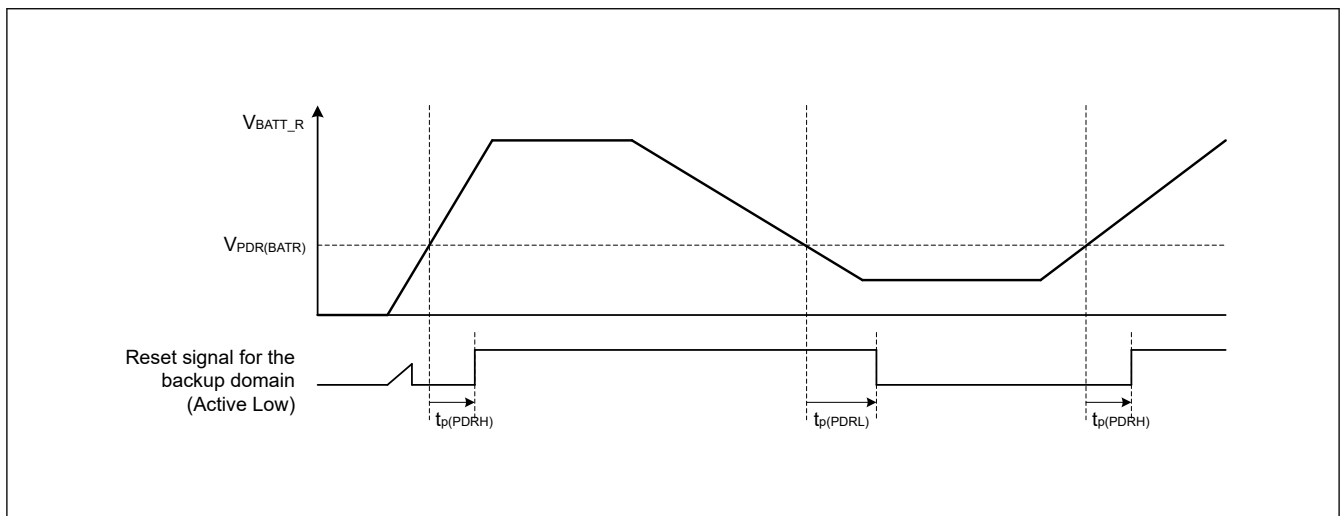


Figure 52.84 Backup Domain Reset Characteristics

### 52.11 ACMPHS Characteristics

Table 52.63 ACMPHS

Parameter	Symbol	Min	Typ	Max	Unit	Test conditions
Reference voltage range	VREF	0	—	AVCC0	V	—
Input voltage range	ACMPHS0 ACMPHS1 IVCMP1 to IVCMP3 IVCMP0	0	—	AVCC0	V	—
		0	—	AVCC0		—
		0	—	AVCC0		VCC ≥ AVCC0
		0	—	VCC		VCC < AVCC0
Output delay*1	Td	—	50	100	ns	VI = VREF ± 100mV
Internal reference voltage	Vref	1.13	1.18	1.28	V	—

Note 1. This value is the internal propagation delay.

## 52.12 Flash Memory Characteristics

## 52.12.1 Code Flash Memory Characteristics

**Table 52.64 Code flash memory characteristics**

Conditions: Program or erase: FCLK = 4 to 60 MHz

Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions	
		Min	Typ* <sup>6</sup>	Max	Min	Typ* <sup>6</sup>	Max			
Programming time N <sub>PEC</sub> ≤ 100 times	128-byte	t <sub>P128</sub>	—	0.75	13.2	—	0.34	6.0	ms	
	8-KB	t <sub>P8K</sub>	—	49	176	—	22	80	ms	
	32-KB	t <sub>P32K</sub>	—	194	704	—	88	320	ms	
Programming time N <sub>PEC</sub> > 100 times	128-byte	t <sub>P128</sub>	—	0.91	15.8	—	0.41	7.2	ms	
	8-KB	t <sub>P8K</sub>	—	60	212	—	27	96	ms	
	32-KB	t <sub>P32K</sub>	—	234	848	—	106	384	ms	
Erasure time N <sub>PEC</sub> ≤ 100 times	8-KB	t <sub>E8K</sub>	—	78	216	—	43	120	ms	
	32-KB	t <sub>E32K</sub>	—	283	864	—	157	480	ms	
Erasure time N <sub>PEC</sub> > 100 times	8-KB	t <sub>E8K</sub>	—	94	260	—	52	144	ms	
	32-KB	t <sub>E32K</sub>	—	341	1040	—	189	576	ms	
Reprogramming/erasure cycle* <sup>4</sup>	N <sub>PEC</sub>	10000* <sup>1</sup>	—	—	—	10000* <sup>1</sup>	—	—	Times	
Suspend delay during programming	t <sub>SPD</sub>	—	—	264	—	—	120	μs		
Programming resume time	t <sub>PRT</sub>	—	—	110	—	—	50	μs		
First suspend delay during erasure in suspend priority mode	t <sub>SESD1</sub>	—	—	216	—	—	120	μs		
Second suspend delay during erasure in suspend priority mode	t <sub>SESD2</sub>	—	—	1.7	—	—	1.7	ms		
Suspend delay during erasure in erasure priority mode	t <sub>SEED</sub>	—	—	1.7	—	—	1.7	ms		
First erasing resume time during erasure in suspend priority mode* <sup>5</sup>	t <sub>REST1</sub>	—	—	1.7	—	—	1.7	ms		
Second erasing resume time during erasure in suspend priority mode	t <sub>REST2</sub>	—	—	144	—	—	80	μs		
Erasing resume time during erasure in erasure priority mode	t <sub>REET</sub>	—	—	144	—	—	80	μs		
Forced stop command	t <sub>FD</sub>	—	—	32	—	—	20	μs		
Data hold time* <sup>2</sup>	t <sub>DRP</sub>	20* <sup>2</sup> * <sup>3</sup>	—	—	—	20* <sup>2</sup> * <sup>3</sup>	—	—	Years	T <sub>j</sub> = +105°C
		30* <sup>2</sup> * <sup>3</sup>	—	—	—	30* <sup>2</sup> * <sup>3</sup>	—	—		T <sub>j</sub> = +85°C

Note 1. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.

Note 2. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.

Note 3. This result is obtained from reliability testing.

Note 4. The reprogram/erase cycle is the number of erasures for each block. When the reprogram/erase cycle is n times (n = 10,000), erasing can be performed n times for each block. For example, when 128-byte programming is performed 64 times for different addresses in 8-KB blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address several times as one erasure is not enabled. Overwriting is prohibited.

Note 5. Time for resumption includes time for reapplying the erasing pulse (up to one full pulse) that was cut off at the time of suspension.

Note 6. The reference value at VCC = 3.3V and room temperature.



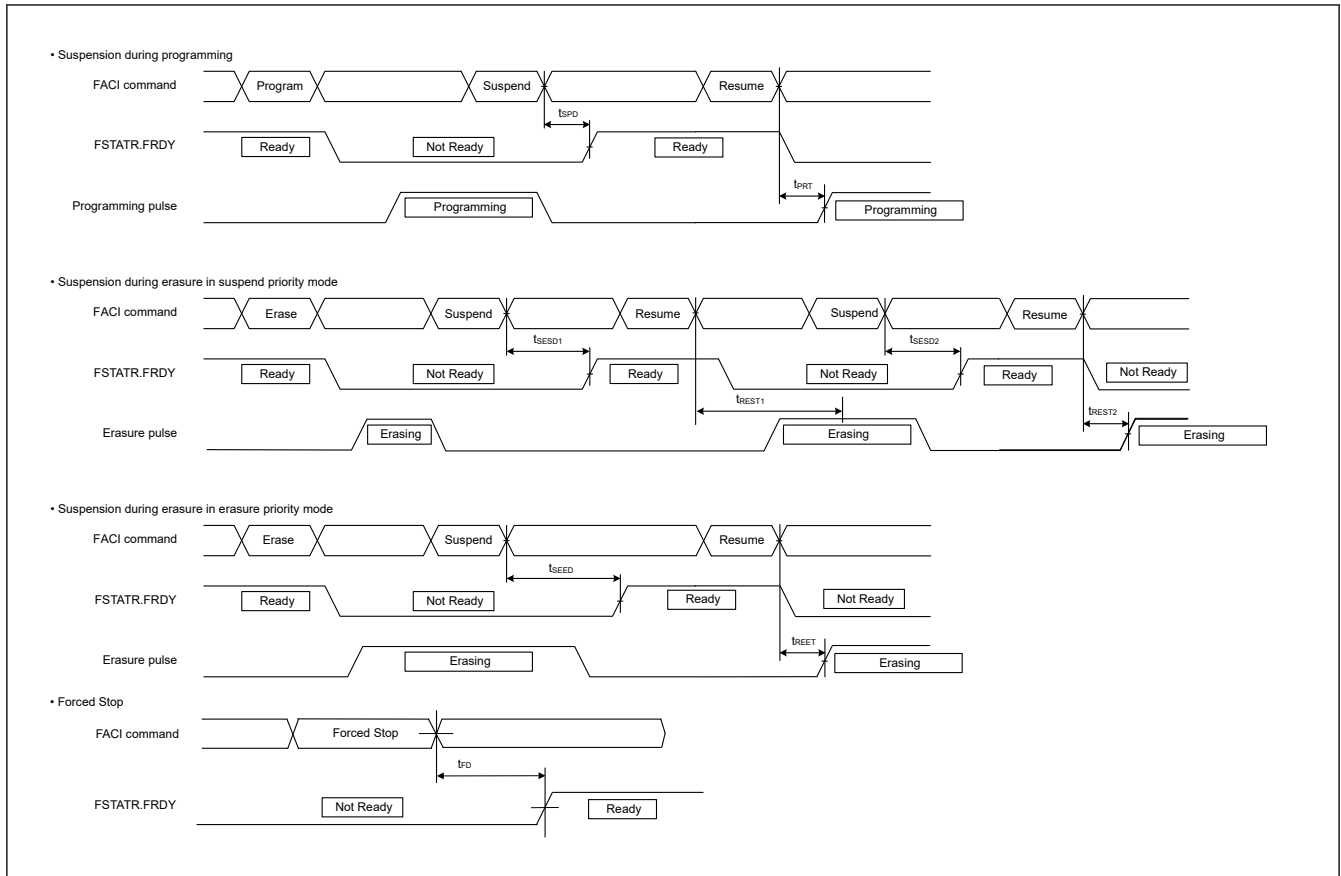


Figure 52.85 Suspension and forced stop timing for flash memory programming and erasure

### 52.12.2 Data Flash Memory Characteristics

Table 52.65 Data flash memory characteristics (1 of 2)

Conditions: Program or erase: FCLK = 4 to 60 MHz  
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
		Min	Typ*6	Max	Min	Typ*6	Max		
Programming time	4-byte	$t_{DP4}$	—	0.36	3.8	—	0.16	1.7	ms
	8-byte	$t_{DP8}$	—	0.38	4.0	—	0.17	1.8	
	16-byte	$t_{DP16}$	—	0.42	4.5	—	0.19	2.0	
Erasure time	64-byte	$t_{DE64}$	—	3.1	18	—	1.7	10	ms
	128-byte	$t_{DE128}$	—	4.7	27	—	2.6	15	
	256-byte	$t_{DE256}$	—	8.9	50	—	4.9	28	
Blank check time	4-byte	$t_{DBC4}$	—	—	84	—	—	30	μs
Reprogramming/erasure cycle*1	$N_{DPEC}$	125000*2	—	—	125000*2	—	—	—	—
Suspend delay during programming	4-byte	$t_{DSPD}$	—	—	264	—	—	120	μs
	8-byte		—	—	264	—	—	120	
	16-byte		—	—	264	—	—	120	
Programming resume time	$t_{DPRT}$	—	—	110	—	—	50	μs	

**Table 52.65 Data flash memory characteristics (2 of 2)**

Conditions: Program or erase: FCLK = 4 to 60 MHz  
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions	
		Min	Typ*6	Max	Min	Typ*6	Max			
First suspend delay during erasure in suspend priority mode	64-byte	t <sub>DSESD1</sub>	—	—	216	—	—	120	μs	
	128-byte		—	—	216	—	—	120		
	256-byte		—	—	216	—	—	120		
Second suspend delay during erasure in suspend priority mode	64-byte	t <sub>DSESD2</sub>	—	—	300	—	—	300	μs	
	128-byte		—	—	390	—	—	390		
	256-byte		—	—	570	—	—	570		
Suspend delay during erasing in erasure priority mode	64-byte	t <sub>DSEED</sub>	—	—	300	—	—	300	μs	
	128-byte		—	—	390	—	—	390		
	256-byte		—	—	570	—	—	570		
First erasing resume time during erasure in suspend priority mode*5	t <sub>DREST1</sub>	—	—	300	—	—	300	μs		
Second erasing resume time during erasure in suspend priority mode	t <sub>DREST2</sub>	—	—	126	—	—	70	μs		
First erasing resume time during erasure in suspend priority mode	t <sub>DREST2</sub>	—	—	126	—	—	70	μs		
Erasing resume time during erasure in erasure priority mode	t <sub>DREET</sub>	—	—	126	—	—	70	μs		
Forced stop command	t <sub>FD</sub>	—	—	32	—	—	20	μs		
Data hold time*3	t <sub>DRP</sub>	20*3 *4	—	—	20*3 *4	—	—	Year	T <sub>j</sub> = +105°C	
		30*3 *4	—	—	30*3 *4	—	—		T <sub>j</sub> = +85°C	

Note 1. The reprogram/erase cycle is the number of erasures for each block. When the reprogram/erase cycle is n times (n = 125,000), erasing can be performed n times for each block. For example, when 4-byte programming is performed 16 times for different addresses in 64-byte blocks, and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address several times as one erasure is not enabled. Overwriting is prohibited.

Note 2. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.

Note 3. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.

Note 4. This result is obtained from reliability testing.

Note 5. Time for resumption includes time for reapplying the erasing pulse (up to one full pulse) that was cut off at the time of suspension.

Note 6. The reference value at VCC = 3.3 V and room temperature.

### 52.12.3 Option Setting Memory (Code flash memory) Characteristics

**Table 52.66 Option setting memory (Code flash memory) characteristics (1 of 2)**

Conditions: Program: FCLK = 4 to 60 MHz  
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
		Min	Typ*4	Max	Min	Typ*4	Max		
Programming time N <sub>OPC</sub> ≤ 200 times	t <sub>OP</sub>	—	83	309	—	45	162	ms	
Programming time N <sub>OPC</sub> > 200 times	t <sub>OP</sub>	—	100	371	—	55	195	ms	

**Table 52.66 Option setting memory (Code flash memory) characteristics (2 of 2)**

Conditions: Program: FCLK = 4 to 60 MHz  
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
		Min	Typ <sup>*4</sup>	Max	Min	Typ <sup>*4</sup>	Max		
Reprogramming cycle	N <sub>OPC</sub>	20000 <sup>*1</sup>	—	—	20000 <sup>*1</sup>	—	—	Times	
Data hold time <sup>*2</sup>	t <sub>DRP</sub>	20 <sup>*2 *3</sup>	—	—	20 <sup>*2 *3</sup>	—	—	Years	T <sub>j</sub> = +105°C
		30 <sup>*2 *3</sup>	—	—	30 <sup>*2 *3</sup>	—	—		T <sub>j</sub> = +85°C

Note 1. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.

Note 2. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.

Note 3. This result is obtained from reliability testing.

Note 4. The reference value at VCC = 3.3 V and room temperature.

## 52.12.4 Option Setting Memory (Data flash memory) Characteristics

**Table 52.67 Option Setting Memory (Data flash memory) characteristics**

Conditions: Program: FCLK = 4 to 60 MHz  
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
		Min	Typ <sup>*4</sup>	Max	Min	Typ <sup>*4</sup>	Max		
Command time for configuration set (4 / 16 Byte)	t <sub>DCCCT</sub>	—	68	515	—	35	255	ms	
Update Cycles in Configuration area	N <sub>cupc</sub>	125000 <sup>*1</sup>	—	—	125000 <sup>*1</sup>	—	—	Times	
Data hold time <sup>*2</sup>	t <sub>DRP</sub>	20 <sup>*2 *3</sup>	—	—	20 <sup>*2 *3</sup>	—	—	Years	T <sub>j</sub> = +105°C
		30 <sup>*2 *3</sup>	—	—	30 <sup>*2 *3</sup>	—	—		T <sub>j</sub> = +85°C

Note 1. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.

Note 2. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.

Note 3. This result is obtained from reliability testing.

Note 4. The reference value at VCC = 3.3 V and room temperature.

## 52.12.5 Anti-rollback counter Characteristics

**Table 52.68 Anti-rollback counter characteristics**

Conditions: Program: FCLK = 4 to 60 MHz  
Read: FCLK ≤ 60 MHz

Parameter	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 60 MHz			Unit	Test conditions
		Min	Typ <sup>*4</sup>	Max	Min	Typ <sup>*4</sup>	Max		
Command time for increment counter and refresh counter	t <sub>IRCT</sub>	—	11.9	81	—	6.3	42	ms	
Command time for read counter	t <sub>RCT</sub>	—	—	25	—	—	5	μs	
Update Cycles (total of increment and refreshing)	N <sub>cupc</sub>	125000 <sup>*1</sup>	—	—	125000 <sup>*1</sup>	—	—	Times	
Data hold time <sup>*2</sup>	t <sub>DRP</sub>	20 <sup>*2 *3</sup>	—	—	20 <sup>*2 *3</sup>	—	—	Years	T <sub>j</sub> = +105°C
		30 <sup>*2 *3</sup>	—	—	30 <sup>*2 *3</sup>	—	—		T <sub>j</sub> = +85°C

Note 1. This is the minimum number of times to guarantee all the characteristics after reprogramming. The guaranteed range is from 1 to the minimum value.

Note 2. This indicates the minimum value of the characteristic when reprogramming is performed within the specified range.

Note 3. This result is obtained from reliability testing.

Note 4. The reference value at VCC = 3.3 V and room temperature.

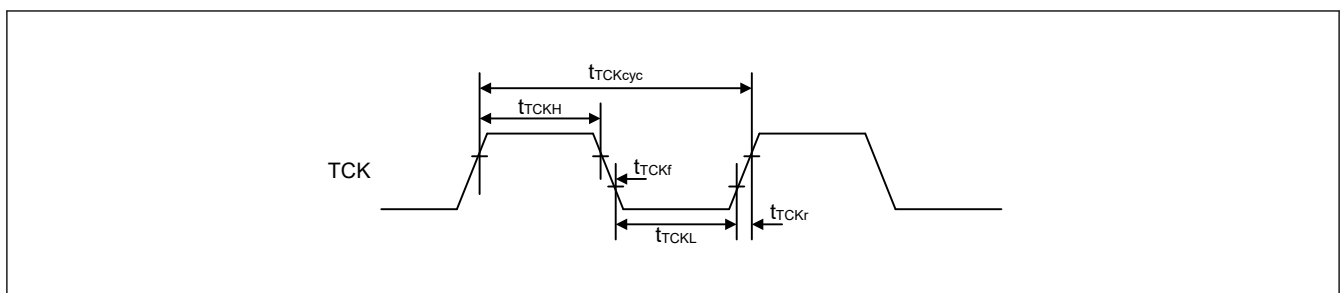
### 52.13 Boundary Scan

**Table 52.69** Boundary scan characteristics

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TCK clock cycle time	1.68 V or above	$t_{TCKcyc}$	100	—	—	ns	Figure 52.86
TCK clock high pulse width	1.68 V or above	$t_{TCKH}$	0.45	—	—	$t_{TCKcyc}$	
TCK clock low pulse width	1.68 V or above	$t_{TCKL}$	0.45	—	—	$t_{TCKcyc}$	
TCK clock rise time	1.68 V or above	$t_{TCKr}$	—	—	$0.05^{*2}$	$t_{TCKcyc}$	
TCK clock fall time	1.68 V or above	$t_{TCKf}$	—	—	$0.05^{*2}$	$t_{TCKcyc}$	
TMS setup time	1.68 V or above	$t_{TMSS}$	20	—	—	ns	Figure 52.87
TMS hold time	1.68 V or above	$t_{TMSh}$	20	—	—	ns	
TDI setup time	1.68 V or above	$t_{TDis}$	20	—	—	ns	
TDI hold time	1.68 V or above	$t_{TDIH}$	20	—	—	ns	
TDO data delay	1.68 V or above	$t_{TDOD}$	—	—	40	ns	
Capture register setup time	1.68 V or above	$t_{CAPTS}$	20	—	—	ns	Figure 52.88
Capture register hold time	1.68 V or above	$t_{CAPTH}$	20	—	—	ns	
Update register delay time	1.68 V or above	$t_{UPDated}$	—	—	40	ns	
Boundary scan circuit startup time <sup>*1</sup>	1.68 V or above	$T_{BSSTUP}$	$t_{RESWP}$	—	—	—	Figure 52.89

Note 1. Boundary scan does not function until the power-on reset becomes negative.

Note 2. 1  $\mu$ s at the longest



**Figure 52.86** Boundary scan TCK timing

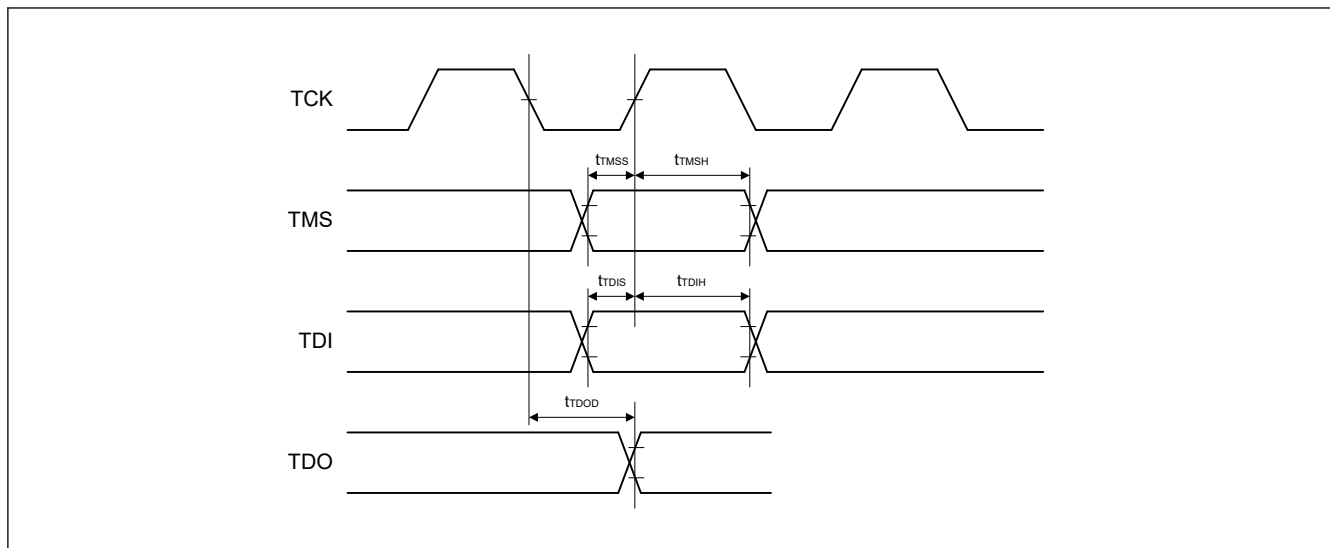


Figure 52.87 Boundary scan input/output timing (1)

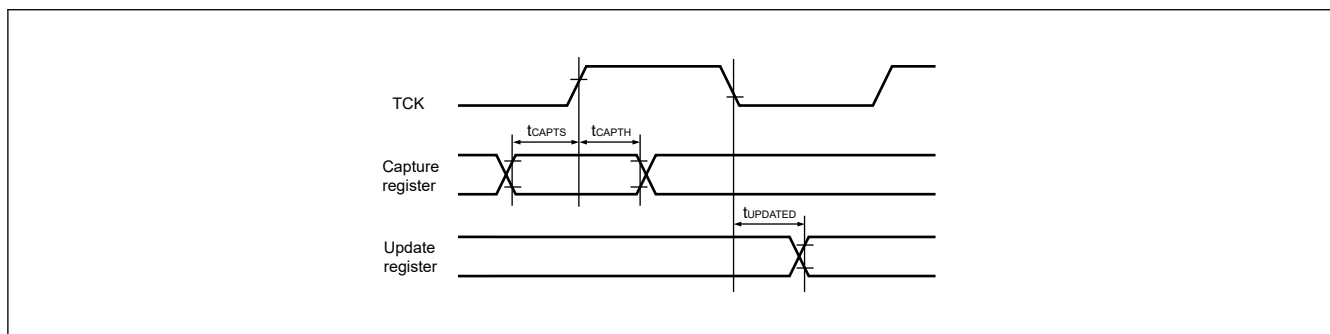


Figure 52.88 Boundary scan input/output timing (2)

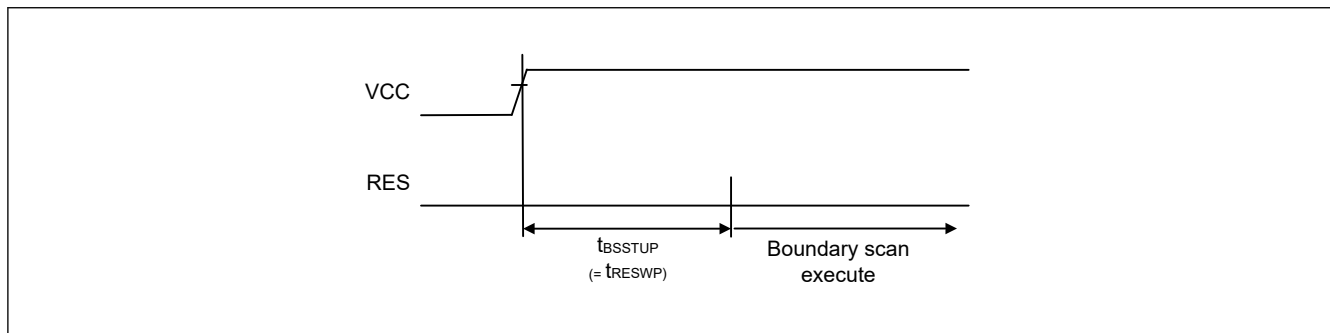


Figure 52.89 Boundary scan circuit startup timing

## 52.14 Joint European Test Action Group (JTAG)

Table 52.70 JTAG

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions	
TCK clock cycle time	2.7 V or above	$t_{TCKcyc}$	40.0	—	—	ns	Figure 52.90	
	1.68 V or above		40.0	—	—	ns		
TCK clock high pulse width	2.7 V or above	$t_{TCKH}$	0.375	—	—	$t_{TCKcyc}$		
	1.68 V or above		0.375	—	—	$t_{TCKcyc}$		
TCK clock low pulse width	2.7 V or above	$t_{TCKL}$	0.375	—	—	$t_{TCKcyc}$		
	1.68 V or above		0.375	—	—	$t_{TCKcyc}$		
TCK clock rise time	2.7 V or above	$t_{TCKr}$	—	—	0.125 <sup>*1</sup>	$t_{TCKcyc}$		
	1.68 V or above		—	—	0.125 <sup>*1</sup>	$t_{TCKcyc}$		
TCK clock fall time	2.7 V or above	$t_{TCKf}$	—	—	0.125 <sup>*1</sup>	$t_{TCKcyc}$		
	1.68 V or above		—	—	0.125 <sup>*1</sup>	$t_{TCKcyc}$		
TMS setup time	2.7 V or above	$t_{TMSS}$	8.0	—	—	ns		Figure 52.91
	1.68 V or above		8.0	—	—	ns		
TMS hold time	2.7 V or above	$t_{TMSH}$	8.0	—	—	ns		
	1.68 V or above		8.0	—	—	ns		
TDI setup time	2.7 V or above	$t_{TDIS}$	8.0	—	—	ns		
	1.68 V or above		8.0	—	—	ns		
TDI hold time	2.7 V or above	$t_{TDIH}$	8.0	—	—	ns		
	1.68 V or above		8.0	—	—	ns		
TDO data delay time	2.7 V or above	$t_{TDOD}$	—	—	20.0	ns		
	1.68 V or above		—	—	28.0	ns		

Note 1. 1  $\mu$ s at the longest

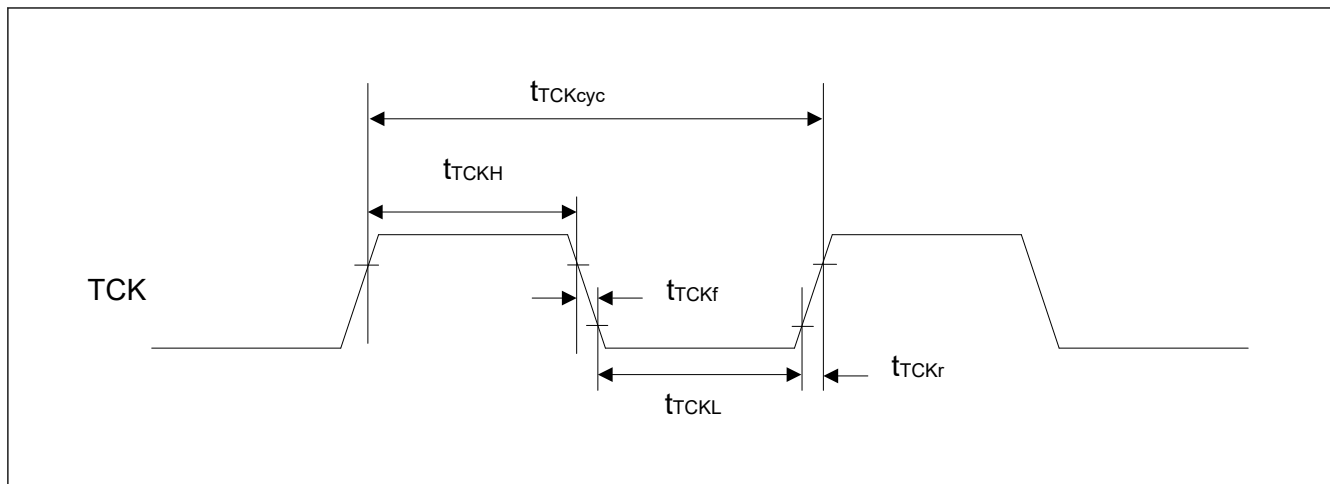


Figure 52.90 JTAG TCK timing

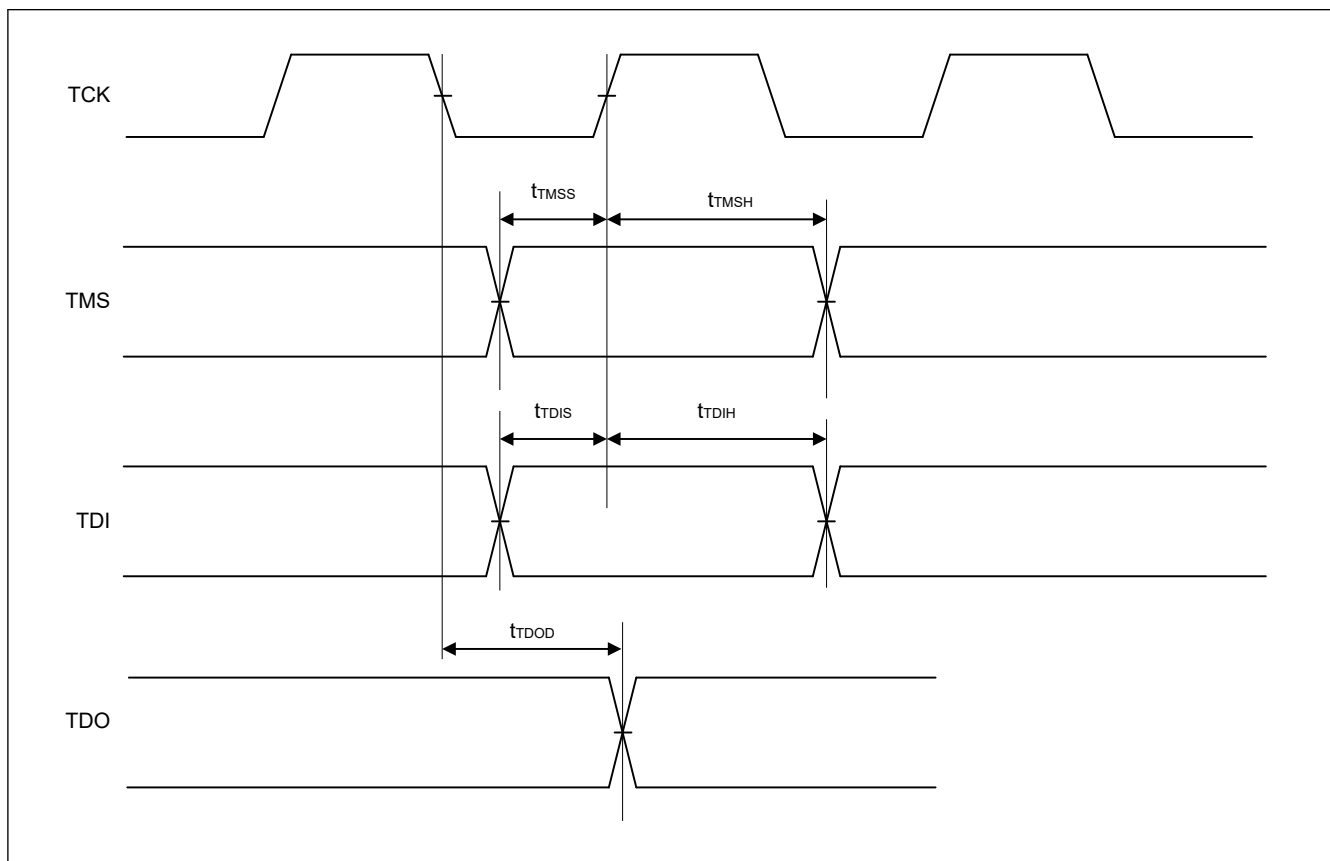


Figure 52.91 JTAG input/output timing

52.15 Serial Wire Debug (SWD)

Table 52.71 SWD

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions	
SWCLK clock cycle time	2.7 V or above	$t_{SWCKcyc}$	40.0	—	—	ns	Figure 52.92	
	1.68 V or above		40.0	—	—	ns		
SWCLK clock high pulse width	2.7 V or above	$t_{SWCKH}$	0.375	—	—	$t_{SWCKcyc}$		
	1.68 V or above		0.375	—	—	$t_{SWCKcyc}$		
SWCLK clock low pulse width	2.7 V or above	$t_{SWCKL}$	0.375	—	—	$t_{SWCKcyc}$		
	1.68 V or above		0.375	—	—	$t_{SWCKcyc}$		
SWCLK clock rise time	2.7 V or above	$t_{SWCKr}$	—	—	$0.125^{*1}$	$t_{SWCKcyc}$		
	1.68 V or above		—	—	$0.125^{*1}$	$t_{SWCKcyc}$		
SWCLK clock fall time	2.7 V or above	$t_{SWCKf}$	—	—	$0.125^{*1}$	$t_{SWCKcyc}$		
	1.68 V or above		—	—	$0.125^{*1}$	$t_{SWCKcyc}$		
SWDIO setup time	2.7 V or above	$t_{SWDS}$	8.0	—	—	ns		Figure 52.93
	1.68 V or above		8.0	—	—	ns		
SWDIO hold time	2.7 V or above	$t_{SWDH}$	8.0	—	—	ns		
	1.68 V or above		8.0	—	—	ns		
SWDIO data delay time	2.7 V or above	$t_{SWDD}$	2.0	—	28.0	ns		
	1.68 V or above		2.0	—	32.0	ns		

Note 1. 1  $\mu$ s at the longest

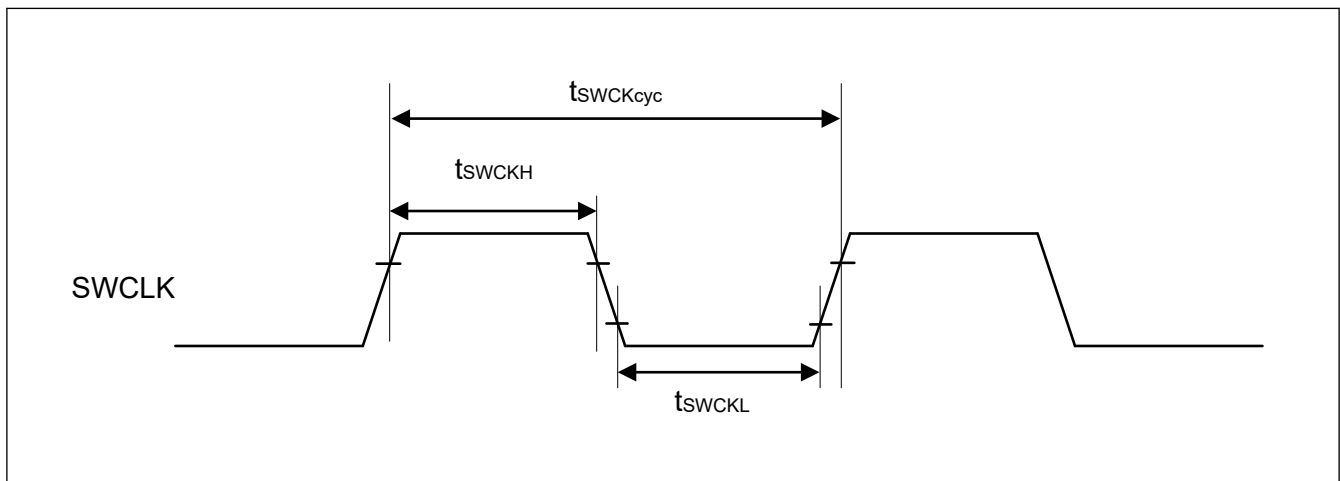


Figure 52.92 SWD SWCLK timing



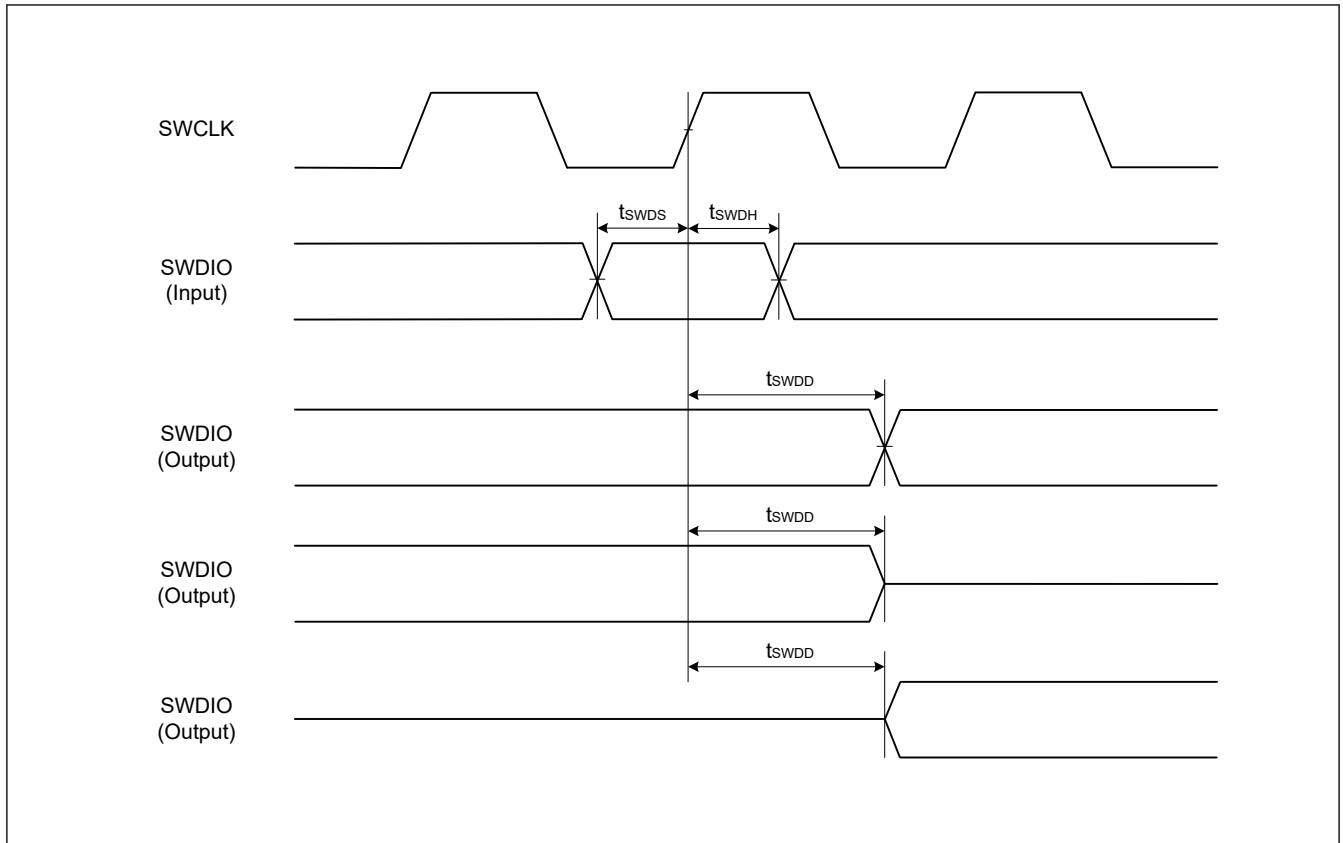


Figure 52.93 SWD input/output timing

### 52.16 Embedded Trace Macro Interface (ETM)

Table 52.72 ETM (1 of 2)

Conditions: High-speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TCLK clock cycle time	2.7 V or above	$t_{TCLKcyc}$	16.6	—	—	ns	Figure 52.94
	1.68 V or above		16.6	—	—	ns	
TCLK clock high pulse width	2.7 V or above	$t_{TCLKH}$	7.3	—	—	ns	
	1.68 V or above		6.3	—	—	ns	
TCLK clock low pulse width	2.7 V or above	$t_{TCLKL}$	7.3	—	—	ns	
	1.68 V or above		6.3	—	—	ns	
TCLK clock rise time	2.7 V or above	$t_{TCLKr}$	—	—	1.0	ns	
	1.68 V or above		—	—	2.0	ns	
TCLK clock fall time	2.7 V or above	$t_{TCLKf}$	—	—	1.0	ns	
	1.68 V or above		—	—	2.0	ns	

**Table 52.72 ETM (2 of 2)**

Conditions: High-speed high drive output is selected in the Port Drive Capability bit in the PmnPFS register.

Parameter	VCC	Symbol	Min	Typ	Max	Unit	Test conditions
TDATA[3:0] output valid time	2.7 V or above	$t_{TRDV}$	—	—	$t_{TCLKcyc}/4 + 1.6$	ns	Figure 52.95
	1.68 V or above		—	—	$t_{TCLKcyc}/4 + 1.6$	ns	
TDATA[3:0] output hold time	2.7 V or above	$t_{TRDH}$	1.5	—	—	ns	
	1.68 V or above		1.5	—	—	ns	

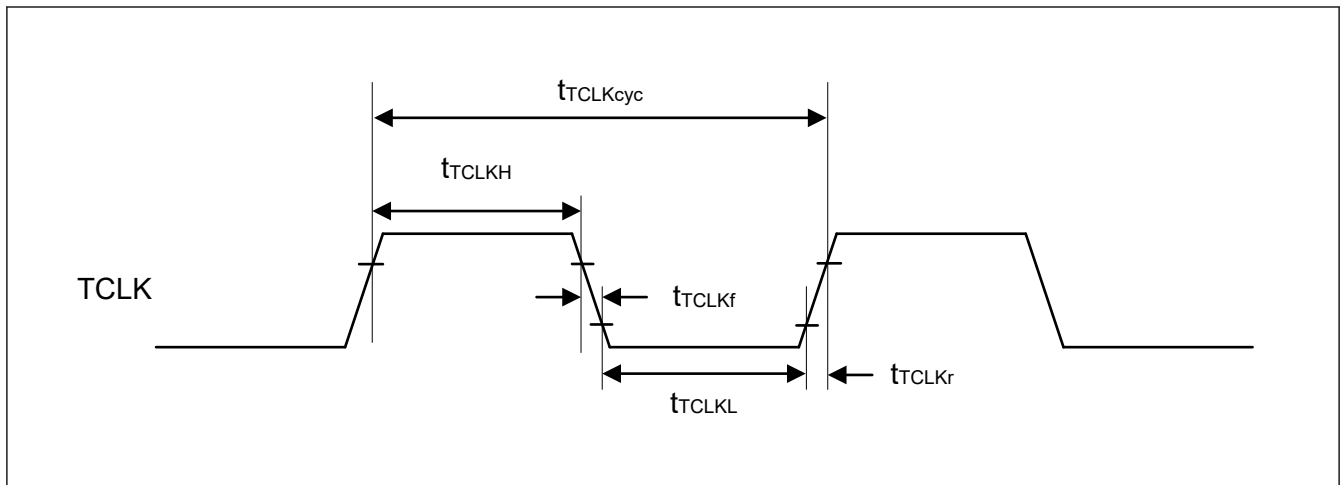


Figure 52.94 ETM TCLK timing

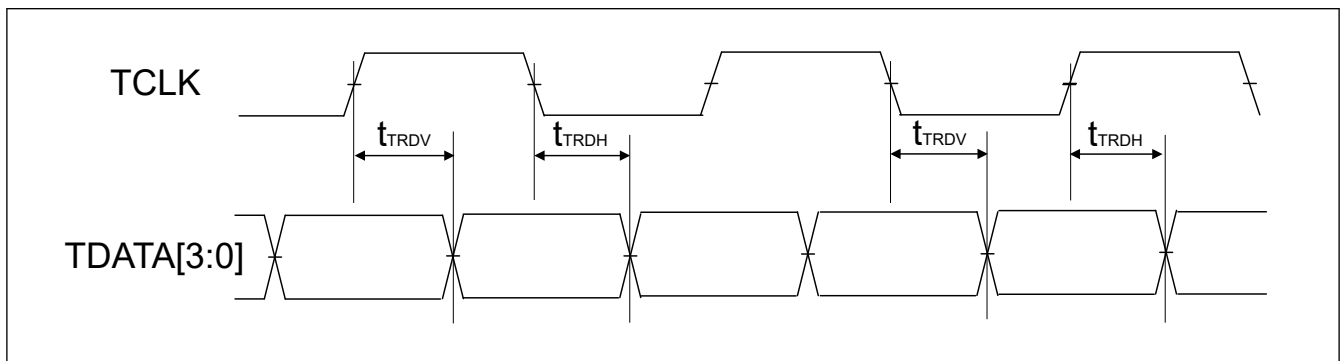


Figure 52.95 ETM output timing

## Appendix 1. Port States in Each Processing Mode

Function	Pin function	Reset	Software Standby mode(SSTBY)		Deep Software Standby mode 1,2,3 (DSTBY1,2,3)		After Deep Software Standby mode is canceled (return to startup mode)	
			OPE=0	OPE=1	DSTBY1	DSTBY2/ DSTBY3	IOKEEP P = 0	IOKEEP = 1 <sup>*1</sup>
Mode	MD	Pull-up	Keep-I		Keep		Pull-up	Keep
JTAG/SWD	TCK/TMS/TDI/SWCLK	Pull-up	TCK/TDI/TMS/SWCLK input		TCK/TDI/TMS/SWCLK input		TCK/TDI/TMS/SWCLK input	
	TDO	Output	TDO output		TDO output		TDO output	
	SWDIO	Pull-up	SWDIO inout		SWDIO inout		SWDIO inout	
Trace	TCLK/TDATAx/SWO	TCLK/ TDATAx/SWO output	TCLK/TDATAx/SWO output		TCLK/TDATAx/SWO output		TCLK/TDATAx/SWO output	
IRQ	IRQx	Hi-Z	Hi-Z <sup>*2</sup>		Keep		Hi-Z	Keep
	IRQx-DS (x:Other than 5)	Hi-Z	Hi-Z <sup>*2</sup>		Keep <sup>*3</sup>		Hi-Z	Keep
	IRQ5-DS	Hi-Z	Hi-Z <sup>*2</sup>		Keep <sup>*3</sup>		Hi-Z	
AGT	AGTIO <sub>n</sub>	Hi-Z	AGTIO <sub>n</sub> inout		Keep		Hi-Z	Keep
	AGTO <sub>n</sub> /AGTOA <sub>n</sub> / AGTOB <sub>n</sub>	Hi-Z	AGTO <sub>n</sub> /AGTOA <sub>n</sub> /AGTOB <sub>n</sub> output		Keep		Hi-Z	Keep
ULPT	ULPTEEn/ULPTEVIn	Hi-Z	ULPTEEn/ULPTEVIn input		Keep		Hi-Z	Keep
	ULPTEEn-DS/ ULPTEVIn-DS	Hi-Z	ULPTEEn-DS/ULPTEVIn-DS input		ULPTEEn-DS/ ULPTEVIn-DS input	Hi-Z	Hi-Z	Keep
	ULPTOn/ ULPTOAn/ ULPTOBn	Hi-Z	ULPTOn/ULPTOAn/ULPTOBn output		Keep		Hi-Z	Keep
	ULPTOn-DS/ ULPTOAn-DS/ ULPTOBn-DS	Hi-Z	ULPTOn/ULPTOAn-DS/ ULPTOBn-DS output		ULPTOn/ ULPTOAn-DS/ ULPTOBn-DS output	Keep	Hi-Z	From DSTBY1: ULPTOn/ ULPTOAn-DS/ ULPTOBn-DS output From DSTBY2,3: Keep
IIC	SCLn/SDAn	Hi-Z	Keep-O <sup>*2</sup>		Keep		Hi-Z	Keep
USBFS	USB_OVRCURx	Hi-Z	Hi-Z <sup>*2</sup>		Keep		Hi-Z	Keep
	USB_OVRCURx-DS/ USB_VBUS	Hi-Z	Hi-Z <sup>*2</sup>		Keep <sup>*3</sup>	Keep	Hi-Z	Keep
	USB_DP/USB_DM	Hi-Z	Keep-O <sup>*4</sup>		Keep <sup>*3</sup>	Keep	Hi-Z	Keep
RTC	RTCICx	Hi-Z	Hi-Z <sup>*2</sup>		Keep <sup>*3</sup>		Hi-Z	Keep
	RTCOUT	Hi-Z	RTCOUT output		Keep		Hi-Z	Keep
ACMPHS	VCOUT	Hi-Z	VCOUT output		Keep		Hi-Z	Keep
CLKOUT	CLKOUT	Hi-Z	CLKOUT output		Keep		Hi-Z	Keep
DAC	DA0	Hi-Z	D/A output retained		Hi-Z		Hi-Z	
P400/P401	Other than function IRQ5-DS	Hi-Z	Keep-O <sup>*2</sup>		Hi-Z		Hi-Z	
Others	—	Hi-Z	Keep-O		Keep		Hi-Z	Keep

Note: Hi-Z: High-impedance

Keep-O: Output pins retain their previous values. Input pins go to high-impedance.

Keep-I: Pin states are retained same as during periods in Normal mode.

Keep: Pin states are retained same as during periods in Software Standby mode.

Note 1. Retains the I/O port state until the DPSBYCR.IOKEEP bit is cleared to 0.

Note 2. Input is enabled if the pin is specified as the Software Standby canceling source while it is used as an external interrupt pin.

Note 3. Input is enabled if the pin is specified as the Deep Software Standby canceling source.

Note 4. Input is enabled while the pin is used as an input pin.

## Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings is displayed in “Packages” on the Renesas Electronics Corporation website.

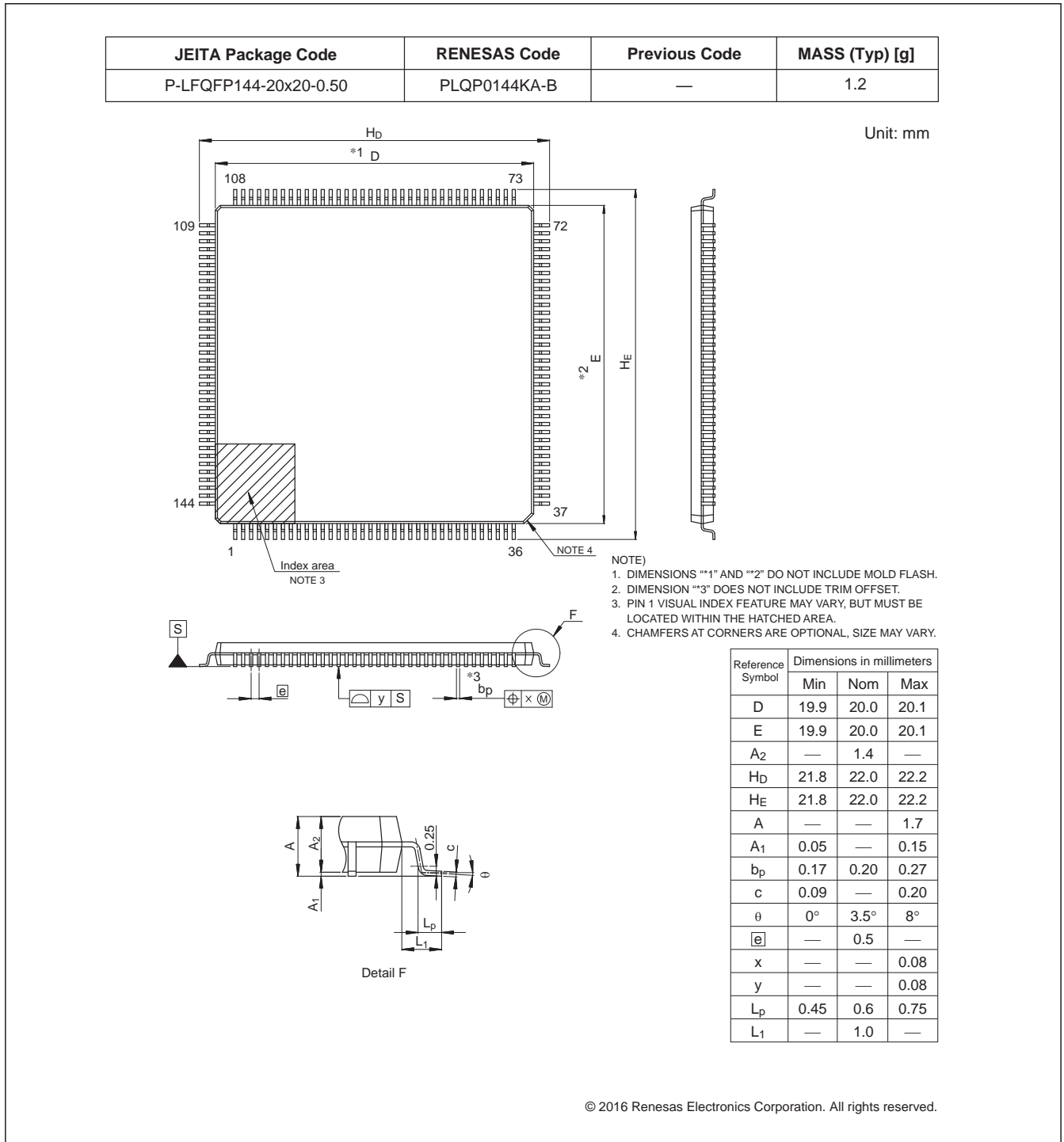
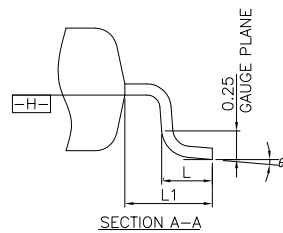
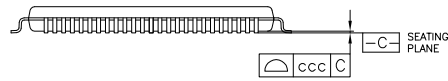
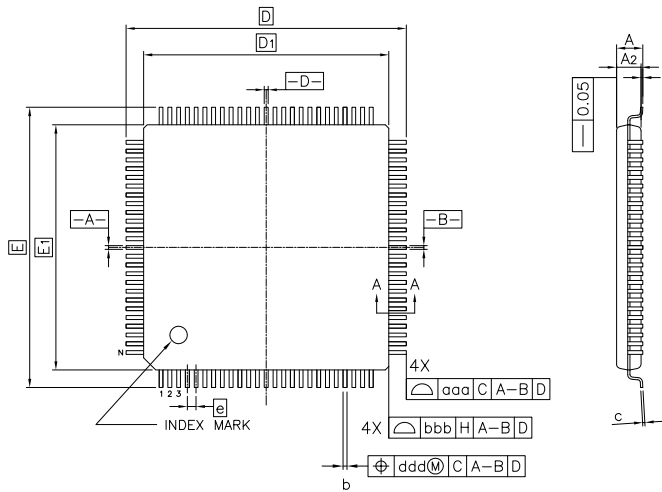


Figure 2.1 LQFP 144-pin

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFQFP100-14x14-0.50	PLQP0100KP-A	0.67



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	1.60
A <sub>1</sub>	0.05	—	0.15
A <sub>2</sub>	1.35	1.40	1.45
D	—	16.00	—
D <sub>1</sub>	—	14.00	—
E	—	16.00	—
E <sub>1</sub>	—	14.00	—
N	—	100	—
e	—	0.50	—
b	0.17	0.22	0.27
c	0.09	—	0.20
θ	0°	3.5°	7°
L	0.45	0.60	0.75
L <sub>1</sub>	—	1.00	—
aaa	—	—	0.20
bbb	—	—	0.20
ccc	—	—	0.08
ddd	—	—	0.08

Figure 2.2 LQFP 100-pin

## Appendix 3. I/O Registers

This appendix describes I/O register address and access cycles by function.

### 3.1 Peripheral Base Addresses

This section provides the base addresses for peripherals described in this manual. [Table 3.1](#) shows the name, description, and the base address of each peripheral.

**Table 3.1 Peripheral base address (1 of 3)**

Description	Name of Secure registers	Base address of Secure registers in Secure alias region	Name of Non-secure registers	Base address of Non-secure registers in Non-secure alias region
Renesas Memory Protection Unit	RMPU	0x4000_0000	RMPU_NS	0x5000_0000
SRAM Control	SRAM	0x4000_2000	SRAM_NS	0x5000_2000
BUS Control	BUS	0x4000_3000	BUS_NS	0x5000_3000
Common Interrupt Controller	ICU_COMMON	0x4000_6000	ICU_COMMON_NS	0x5000_6000
CPU System Security Control Unit	CPSCU	0x4000_8000	CPSCU_NS	0x5000_8000
Direct memory access controller 00	DMAC00	0x4000_A000	DMAC00_NS	0x5000_A000
Direct memory access controller 01	DMAC01	0x4000_A040	DMAC01_NS	0x5000_A040
Direct memory access controller 02	DMAC02	0x4000_A080	DMAC02_NS	0x5000_A080
Direct memory access controller 03	DMAC03	0x4000_A0C0	DMAC03_NS	0x5000_A0C0
Direct memory access controller 04	DMAC04	0x4000_A100	DMAC04_NS	0x5000_A100
Direct memory access controller 05	DMAC05	0x4000_A140	DMAC05_NS	0x5000_A140
Direct memory access controller 06	DMAC06	0x4000_A180	DMAC06_NS	0x5000_A180
Direct memory access controller 07	DMAC07	0x4000_A1C0	DMAC07_NS	0x5000_A1C0
DMAC Module Activation 0	DMA0	0x4000_A800	DMA0_NS	0x5000_A800
Data Transfer Controller 0	DTC0	0x4000_AC00	DTC0_NS	0x5000_AC00
Interrupt Controller	ICU	0x4000_C000	ICU_NS	0x5000_C000
CPU Control Registers	CPU_CTRL	0x4000_F000	CPU_CTRL_NS	0x5000_F000
On-Chip Debug	OCD_CPU	0x4001_1000	OCD_CPU_NS	0x5001_1000
DAP Function	DAP_CPU	0x8001_1000		
Debug Function	CPU_DBG	0x4001_B000	CPU_DBG_NS	0x5001_B000
System Control	SYSC	0x4001_E000	SYSC_NS	0x5001_E000
Temperature Sensor Data	TSD	0x4011_B000	TSD_NS	0x5011_B000
Event Link Controller	ELC	0x4020_1000	ELC_NS	0x5020_1000
Realtime Clock	RTC	0x4020_2000	RTC_NS	0x5020_2000
Independent Watchdog Timer	IWDT	0x4020_2200	IWDT_NS	0x5020_2200
Clock Frequency Accuracy Measurement Circuit	CAC	0x4020_2400	CAC_NS	0x5020_2400
Watchdog Timer 0	WDT0	0x4020_2600	WDT0_NS	0x5020_2600
Module Stop Control A,B,C,D,E	MSTP	0x4020_3000	MSTP_NS	0x5020_3000
Peripheral Security Control Unit	PSCU	0x4020_4000	PSCU_NS	0x5020_4000
Port Output Enable Module for GPT	POEG	0x4021_2000	POEG_NS	0x5021_2000
Ultra-Low Power Timer 0	ULPT0	0x4022_0000	ULPT0_NS	0x5022_0000
Ultra-Low Power Timer 1	ULPT1	0x4022_0100	ULPT1_NS	0x5022_0100

**Table 3.1 Peripheral base address (2 of 3)**

Description	Name of Secure registers	Base address of Secure registers in Secure alias region	Name of Non-secure registers	Base address of Non-secure registers in Non-secure alias region
Low Power Asynchronous General purpose Timer 0	AGT0	0x4022_1000	AGT0_NS	0x5022_1000
Low Power Asynchronous General purpose Timer 1	AGT1	0x4022_1100	AGT1_NS	0x5022_1100
Temperature Sensor	TSN	0x4023_5000	TSN_NS	0x5023_5000
High-Speed Analog Comparator 0	ACMPHS0	0x4023_6000	ACMPHS0_NS	0x5023_6000
High-Speed Analog Comparator 1	ACMPHS1	0x4023_6100	ACMPHS1_NS	0x5023_6100
USB 2.0 FS Module	USBFS	0x4025_0000	USBFS_NS	0x5025_0000
Serial Sound Interface Enhanced (SSIE) 0	SSIE0	0x4025_D000	SSIE0_NS	0x5025_D000
Serial Sound Interface Enhanced (SSIE) 1	SSIE1	0x4025_D100	SSIE1_NS	0x5025_D100
Inter-Integrated Circuit 0	IIC0	0x4025_E000	IIC0_NS	0x5025_E000
Inter-Integrated Circuit 0 Wake-up Unit	IIC0WU	0x4025_E014	IIC0WU_NS	0x5025_E014
Inter-Integrated Circuit 1	IIC1	0x4025_E100	IIC1_NS	0x5025_E100
Octal Serial Peripheral Interface 0	OSPI0_B	0x4026_8000	OSPI0_B_NS	0x5026_8000
CRC Calculator	CRC	0x4031_0000	CRC_NS	0x5031_0000
Data Operation Circuit	DOC_B	0x4031_1000	DOC_B_NS	0x5031_1000
General PWM 32-bit Timer 0	GPT320	0x4032_2000	GPT320_NS	0x5032_2000
General PWM 32-bit Timer 1	GPT321	0x4032_2100	GPT321_NS	0x5032_2100
General PWM 32-bit Timer 2	GPT322	0x4032_2200	GPT322_NS	0x5032_2200
General PWM 32-bit Timer 3	GPT323	0x4032_2300	GPT323_NS	0x5032_2300
General PWM 32-bit Timer 4	GPT324	0x4032_2400	GPT324_NS	0x5032_2400
General PWM 32-bit Timer 5	GPT325	0x4032_2500	GPT325_NS	0x5032_2500
General PWM 16-bit Timer 10	GPT1610	0x4032_2A00	GPT1610_NS	0x5032_2A00
General PWM 16-bit Timer 11	GPT1611	0x4032_2B00	GPT1611_NS	0x5032_2B00
General PWM 16-bit Timer 12	GPT1612	0x4032_2C00	GPT1612_NS	0x5032_2C00
General PWM 16-bit Timer 13	GPT1613	0x4032_2D00	GPT1613_NS	0x5032_2D00
12bit A/D Converter 0	ADC120	0x4033_2000	ADC120_NS	0x5033_2000
12bit A/D Converter 1	ADC121	0x4033_2200	ADC121_NS	0x5033_2200
12-bit D/A converter	DAC12	0x4033_3000	DAC12_NS	0x5033_3000
Capture Engine Unit	CEU	0x4034_8000	CEU_NS	0x5034_8000
DMA Controller for the Ethernet Controller Channel 0	EDMAC0	0x4035_4000	EDMAC0_NS	0x5035_4000
Ethernet Controller Channel 0	ETHERC0	0x4035_4100	ETHERC0_NS	0x5035_4100
Serial Communication Interface 0	SCI0_B	0x4035_8000	SCI0_B_NS	0x5035_8000
Serial Communication Interface 1	SCI1_B	0x4035_8100	SCI1_B_NS	0x5035_8100
Serial Communication Interface 2	SCI2_B	0x4035_8200	SCI2_B_NS	0x5035_8200
Serial Communication Interface 3	SCI3_B	0x4035_8300	SCI3_B_NS	0x5035_8300
Serial Communication Interface 4	SCI4_B	0x4035_8400	SCI4_B_NS	0x5035_8400
Serial Communication Interface 9	SCI9_B	0x4035_8900	SCI9_B_NS	0x5035_8900

**Table 3.1 Peripheral base address (3 of 3)**

Description	Name of Secure registers	Base address of Secure registers in Secure alias region	Name of Non-secure registers	Base address of Non-secure registers in Non-secure alias region
Serial Peripheral Interface 0	SPI0	0x4035_C000	SPI0_NS	0x5035_C000
Serial Peripheral Interface 1	SPI1	0x4035_C100	SPI1_NS	0x5035_C100
Error correction circuit for MBRAM0	ECCMB0	0x4036_F200	ECCMB0_NS	0x5036_F200
Error correction circuit for MBRAM1	ECCMB1	0x4036_F300	ECCMB1_NS	0x5036_F300
CANFD Module 0	CANFD0	0x4038_0000	CANFD0_NS	0x5038_0000
CANFD Module 1	CANFD1	0x4038_2000	CANFD1_NS	0x5038_2000
Port 0 Control Registers	PORT0	0x4040_0000	PORT0_NS	0x5040_0000
Port 1 Control Registers	PORT1	0x4040_0020	PORT1_NS	0x5040_0020
Port 2 Control Registers	PORT2	0x4040_0040	PORT2_NS	0x5040_0040
Port 3 Control Registers	PORT3	0x4040_0060	PORT3_NS	0x5040_0060
Port 4 Control Registers	PORT4	0x4040_0080	PORT4_NS	0x5040_0080
Port 5 Control Registers	PORT5	0x4040_00A0	PORT5_NS	0x5040_00A0
Port 6 Control Registers	PORT6	0x4040_00C0	PORT6_NS	0x5040_00C0
Port 7 Control Registers	PORT7	0x4040_00E0	PORT7_NS	0x5040_00E0
Port 8 Control Registers	PORT8	0x4040_0100	PORT8_NS	0x5040_0100
Port 9 Control Registers	PORT9	0x4040_0120	PORT9_NS	0x5040_0120
Pmn Pin Function Control Register	PFS	0x4040_0800	PFS_NS	0x5040_0800
Flash Cache	FCACHE	0x4001_C100	FCACHE_NS	0x5001_C100
Data Flash	FLAD	0x4011_C000	FLAD_NS	0x5011_C000
Flash Application Command Interface	FACI	0x4011_E000	FACI_NS	0x5011_E000
Data Flash Security Setting	FDFS	0x2703_0000		

Note: Name = Peripheral name  
Description = Peripheral functionality  
Base address = Lowest reserved address or address used by the peripheral

## 3.2 Access Cycles

This section provides access cycle information for the I/O registers described in this manual.

- Registers are grouped by associated module.
- The number of access cycles indicates the number of cycles based on the specified reference clock.
- In the internal I/O area, reserved addresses that are not allocated to registers must not be accessed, otherwise operations cannot be guaranteed.
- The number of I/O access cycles depends on bus cycles of the internal peripheral bus, divided clock synchronization cycles, and wait cycles of each module. Divided clock synchronization cycles differ depending on the frequency ratio between ICLK and PCLK.
- When the frequency of ICLK is equal to that of PCLK, the number of divided clock synchronization cycles is always constant.
- When the frequency of ICLK is greater than that of PCLK, at least 1 PCLK cycle is added to the number of divided clock synchronization cycles.
- The number of write access cycles indicates the number of cycles obtained by non-bufferable write access.

Note: This applies to the number of cycles when access from the CPU does not conflict with the instruction fetching to the external memory or bus access from other bus masters such as DTC or DMAC.



**Table 3.2 Access cycles (1 of 3)**

Peripheral base address symbol	Address <sup>*1</sup>		Number of access cycles				Cycle Unit	Related function
			ICLK = PCLK		ICLK > PCLK <sup>*2</sup>			
	From	To	Read	Write	Read	Write		
RMPU, SRAM, BUS, ICU_COMMON, CPSCU, DMAC0n, DMA0, DTC0, ICU, CPU_CTRL	0x4000_0000	0x4001_CFFF	3	2	3	2	ICLK	Renesas Memory Protection Unit, SRAM Control, BUS Control, Common Interrupt Controller, CPU System Security Control Unit, Direct memory access controller 0 n, DMAC Module Activation 0, Data Transfer Controller 0, Interrupt Controller, CPU Control Registers
CPU_OCD	0x4001_1004	0x4001_1FFF	7	2	7	2	ICLK	On-Chip Debug
CPU_DBG, FCACHE	0x4000_B000	0x4001_CFFF	3	2	3	2	ICLK	Debug Function, Flash Cache
SYSC	0x4001_E000	0x4001_E9FF	4	3	2 to 4	1 to 3	PCLK B	System Control
SYSC	0x4001_EA00	0x4001_ED7F	7	6	5 to 7	4 to 6	PCLK B	System Control
TSD	0x4011_B17C	0x4011_B17C	4	3	4	3	ICLK	Temperature Sensor Data
ELC, RTC	0x4020_1000	0x4020_21FF	4	3	2 to 4	1 to 3	PCLK B	Event Link Controller, Realtime Clock
IWDT	0x4020_2200	0x4020_22FF	4	65	2 to 4	63 to 65	PCLK B	Independent Watchdog Timer
CAC, WDT0, MSTP, PSCU, POEG	0x4020_2400	0x4021_2FFF	4	3	2 to 4	1 to 3	PCLK B	Clock Frequency Accuracy Measurement Circuit, Watchdog Timer 0, Module Stop Control, Peripheral Security Control Unit, Port Output Enable Module for GPT
ULPTn	0x4022_0000	0x4022_01FF	6	65	4 to 6	63 to 65	PCLK B	Ultra-Low Power Timer n
AGTn	0x4022_1000	0x4022_11FF	6	3	4 to 6	1 to 3	PCLK B	Low Power Asynchronous General purpose Timer n
TSN	0x4023_5000	0x4023_5FFF	4	3	2 to 4	1 to 3	PCLK B	Temperature Sensor
ACMPHSn	0x4023_6000	0x4023_61FF	3	3	1 to 3	1 to 3	PCLK B	High-Speed Analog Comparator n
USBFS	0x4025_0000	0x4025_03FF	5	4	3 to 5	2 to 4	PCLK B	USB 2.0 FS Module
USBFS	0x4025_0400	0x4025_04FF	4	65	2 to 4	63 to 65	PCLK B	USB 2.0 FS Module
SSIEn, IICn, OSPIO	0x4025_2000	0x4026_88FF	4	3	2 to 4	1 to 3	PCLK B	Serial Sound Interface Enhanced n, Inter-Integrated Circuit n, Octal Serial Peripheral Interface 0
CRC, DOC	0x4031_0000	0x4031_1FFF	4	3	2 to 4	1 to 3	PCLK A	CRC Calculator, Data Operation Circuit
GPT32n, GPT16n	0x4032_2000	0x4032_3FFF	7	4	5 to 7	2 to 4	PCLK A	General PWM 32-Bit Timer n, General PWM 16-Bit Timer n
ADC12n, DAC12n	0x4033_2000	0x4034_6FFF	4	3	2 to 4	1 to 3	PCLK A	12-bit A/D Converter n, 12-bit D/A Converter n
CEU	0x4034_8000	0x4034_FFFF	7	5	5 to 7	3 to 5	PCLK A	Capture Engine Unit
EDMAC0	0x4035_4000	0x4035_40FF	5	4	3 to 5	2 to 4	PCLK A	DMA Controller for the Ethernet Controller Channel 0
ETHERC0	0x4035_4100	0x4035_43FF	14	13	12 to 14	11 to 13	PCLK A	Ethernet Controller Channel 0
SCIn, SPIn	0x4035_8000	0x4035_FFFF	4	3	2 to 4	1 to 3	PCLK A	Serial Communication Interface n, Serial Peripheral Interface n
ECCMBn	0x4036_F200	0x4036_F3FF	5	4	3 to 5	2 to 4	PCLK A	Error correction circuit for MBRAMn
CANFDn	0x4038_0000	0x4038_3FFF	4	3	2 to 4	1 to 3	PCLK A	CANFD Module n

**Table 3.2 Access cycles (2 of 3)**

Peripheral base address symbol	Address* <sup>1</sup>		Number of access cycles					Cycle Unit	Related function
			ICLK = PCLK		ICLK > PCLK* <sup>2</sup>				
	From	To	Read	Write	Read	Write			
PORTn	0x4040_0000	0x4040_01FF	4	2	4	2	ICLK	Port n Control Registers	
PFS	0x4040_0800	0x4040_0FFF	8	2	8	2	ICLK	Pmn Pin Function Control Register	
RSIP-E51A	—	—	1 to 3	2	1 to 3	1 to 2	PCLK A	Renesas Security IP	

**Table 3.2 Access cycles (3 of 3)**

Peripheral base address symbol	Address* <sup>1</sup>		Number of access cycles				Cycle Unit	Related function
			ICLK = FCLK		ICLK > FCLK* <sup>2</sup>			
	From	To	Read	Write	Read	Write		
FLAD, FACL	0x4011_C040	0x4011_EFFF	4	3	4	3	FCLK	Data Flash, Flash Application Command Interface

Note 1. This table only shows secure address. Access cycle of the non-secure address is the same as its secure address.

Note 2. If the number of PCLK or FCLK cycles is non-integer (for example 1.5), the minimum value is without the decimal point, and the maximum value is rounded up to the decimal point. For example, 1.5 to 2.5 is 1 to 3.

## Appendix 4. Note for Register R/W

- A secure bus master issues a "secure access" using an address marked as secure by IDAU/SAU or MSAU.
- A secure bus master issues a "non-secure access" using an address marked as non-secure by IDAU/SAU or MSAU.
- A non-secure bus master issues a "non-secure access" using an address marked as non-secure by IDAU/SAU or MSAU.

**Table 4.1 Type of Register Notes(S-TYPE)**

TYPE	UM Description
S-TYPE-1	Only Secure access can write to this register. Read access is always allowed. Non-secure write access is ignored, but TrustZone access error is not generated.
S-TYPE-2	Read access is always allowed If the security attribution is configured as Secure, <ul style="list-style-type: none"> <li>• Secure write access is allowed.</li> <li>• Non-secure write access is ignored, but TrustZone access error is not generated.</li> </ul>
	If the security attribution is configured as Non-secure, <ul style="list-style-type: none"> <li>• Secure write access is ignored, but TrustZone access error is not generated.</li> <li>• Non-secure access is allowed.</li> </ul>
S-TYPE-3	If the security attribution is configured as Secure, <ul style="list-style-type: none"> <li>• Secure access is allowed.</li> <li>• Non-secure write access is ignored and Non-secure read access is read as 0, TrustZone access error is generated</li> </ul>
	If the security attribution is configured as Non-secure, <ul style="list-style-type: none"> <li>• Secure write access is ignored and Secure read access is read as 0, TrustZone access error is generated.</li> <li>• Non-secure access is allowed</li> </ul>
S-TYPE-4	If the security attribution is configured as Secure, <ul style="list-style-type: none"> <li>• Secure access is allowed</li> <li>• Non-secure write access is ignored and Non-secure read access is read as 0, but TrustZone access error is not generated.</li> </ul>
	If the security attribution is configured as Non-secure, <ul style="list-style-type: none"> <li>• Secure write access is ignored and Secure read access is read as 0, but TrustZone access error is not generated.</li> <li>• Non-secure access is allowed.</li> </ul>
S-TYPE-5	No note required.
S-TYPE-6	Secure access is allowed. Non-secure write access is ignored, and Non-secure read access is read as 0, TrustZone access error is generated.
S-TYPE-7	Secure write access is ignored, and Secure read access is read as 0, TrustZone access error is generated. Non-secure access is allowed.

Note: A non-secure bus master does NOT issue any access using an address marked as secure by IDAU/SAU or MSAU.

**Table 4.2 Type of Register Notes(P-TYPE)**

TYPE	UM Description
P-TYPE-1	Privileged write access is allowed. Read access is always allowed. Unprivileged write access is ignored, but TrustZone access error is not generated.
P-TYPE-2	Privileged access is allowed. Unprivileged write access is ignored, and Unprivileged read access is read as 0, TrustZone access error is generated.
P-TYPE-3	If the privilege attribution is configured as Privileged, <ul style="list-style-type: none"> <li>• Privileged access is allowed.</li> <li>• Unprivileged write access is ignored and Unprivileged read access is read as 0, TrustZone access error is generated.</li> </ul>
	If the privilege attribution is configured as Unprivilege, <ul style="list-style-type: none"> <li>• Privileged access and Unprivileged access are allowed.</li> </ul>
P-TYPE-4	If the privilege attribution is configured as Privileged, <ul style="list-style-type: none"> <li>• Privileged access is allowed.</li> <li>• Unprivileged write access is ignored and Unprivileged read access is read as 0, TrustZone access error is not generated.</li> </ul>
	If the privilege attribution is configured as Unprivilege, <ul style="list-style-type: none"> <li>• Privileged access and Unprivileged access are allowed.</li> </ul>
P-TYPE-5	No note required.

## Appendix 5. Peripheral Variant

Table 5.1 shows the correspondence between the module name used in this manual and the Peripheral Variant.

**Table 5.1 Module name vs Peripheral Variant**

Module name	Peripheral variant
SCI	SCI_B
SPI	SPI_B
OSPI	OSPI_B

---

## Revision History

### Revision 1.00 — October 23, 2024

Initial release

---

RA8E1 Group User's Manual: Hardware

Publication Date: Rev.1.00 Oct 23, 2024

Published by: Renesas Electronics Corporation

---

32-Bit MCU  
RA8E1 Group